Outils de calcul quantique tolérant aux fautes

par

Guillaume Duclos-Cianci

Thèse présentée au département de physique
en vue de l'obtention du grade de docteur ès sciences (Ph.D.)

FACULTÉ des SCIENCES
UNIVERSITÉ de SHERBROOKE

Sherbrooke, Québec, Canada, 24 avril 2015

Le 24 avril 2015

le jury a accepté la thèse de Monsieur Guillaume Duclos-Cianci dans sa version finale.

Membres du jury

Professeur David Poulin
Directeur de recherche
Département de physique

Professeur Patrick Fournier
Membre interne
Département de physique

Professor Robert Raussendorf
Membre externe
Department of Physics and Astronomy
University of British Columbia

Professeur David Sénéchal
Président rapporteur
Département de physique

À mes parents et amis

Sommaire

Le développement de qubits quantiques robustes représente un défi technologique de taille. Malgré plus d'une décennie de progrès et de percées, nous sommes toujours à la recherche du candidat idéal. La difficulté réside dans la nécessité de respecter une panoplie de critères stricts : on doit pouvoir préparer et mesurer les qubits rapidement et de manière fiable, préserver leur état pour de longs temps, appliquer avec précision un continuum de transformations, les coupler les uns aux autres, en entasser des milliers, voire des millions sur un seul dispositif, etc.

Parallèlement à ces recherches, un autre groupe de scientifiques travaillent plutôt à l'élaboration de l'architecture permettant d'opérer ces qubits. Cette architecture inclut une couche logicielle de base dont l'étude constitue le domaine du calcul tolérant aux fautes : en encodant l'information dans des qubits logiques à l'aide des qubits physiques disponibles, il est possible d'obtenir un dispositif quantique dont les propriétés effectives sont supérieures à celles des composantes physiques sous-jacentes. En contrepartie, une surcharge doit être payée. Celle-ci peut être interprétée comme une forme de redondance dans l'information. De plus, les portes logiques applicables aux qubits encodés sont souvent trop limitées pour être utiles. La recherche dans ce domaine vise souvent à limiter la surcharge et à étendre l'ensemble des opérations applicables.

Cette thèse présente les travaux que j'ai publiés avec mes collaborateurs durant mes études de doctorat. Ceux-ci touchent deux aspects importants du calcul tolérant aux fautes : l'élaboration de protocoles de calcul universel et la conception et l'étude d'algorithmes de décodage de codes topologiques stabilisateurs.

Concernant l'élaboration de protocoles de calcul universel, j'ai développé avec l'aide de Krysta Svore chez Microsoft Research une nouvelle famille d'états ressources (Chapitre 2). Celle-ci permet, par l'injection d'états, d'effectuer une opération unitaire arbitraire à un qubit à un coût plus faible que les méthodes existant à ce moment. Plus tard, j'ai poursuivi ces travaux avec David Poulin pour élaborer une autre famille d'états ressources qui diminuent

encore davantage les coûts de compilation de diverses portes unitaires à un qubit (Chapitre 3). Finalement, Jonas Anderson, David Poulin et moi avons montré comment il est possible de passer de manière tolérante aux fautes d'un encodage à un autre (Chapitre 4). Cette approche est qualitativement différente, car elle fournit un ensemble universel de portes sans passer par l'injection d'états.

Durant mon doctorat, j'ai aussi généralisé de plusieurs manières la méthode de décodage par renormalisation du code topologique de Kitaev que j'ai développée au cours de ma maîtrise. Tout d'abord, j'ai collaboré avec Héctor Bombin et David Poulin dans le but de montrer que tous les codes topologiques stabilisateurs invariants sous translation sont équivalents, c'est-à-dire qu'ils appartiennent à la même phase topologique (Chapitre 5). Ce résultat m'a aussi permis d'adapter mon décodeur aux codes topologiques de couleurs stabilisateurs et à sous-systèmes. Puis, je l'ai adapté à une généralisation du code topologique de Kitaev sur des qudits (Chapitre 6). Ensuite, je l'ai généralisé au cas tolérant aux fautes, où les erreurs dans les mesures du syndrome sont prises en compte (Chapitre 7). Finalement, je l'ai appliqué à un nouveau code élaboré par Sergey Bravyi, le code de surface à sous-systèmes (Chapitre 8).

Mots-clés : calcul quantique tolérant aux fautes, ensemble universel de portes, distillation d'états magiques, déformation de codes, équivalence de codes topologiques stabilisateurs, décodage de codes topologiques stabilisateurs et à sous-systèmes.

Remerciements

J'aimerais remercier ma famille qui m'a toujours soutenu et qui a été très influente dans ma vie. Merci de m'avoir inculqué mes valeurs qui me soutiennent et me permettent de persévérer dans mon travail.

Merci à ma conjointe qui a démontré un support et une patience exemplaires durant les moments éprouvants qu'amène inévitablement un projet de cette envergure.

Merci à mes frères d'armes qui se reconnaîtront et avec qui j'ai partagé les hauts et les bas du quotidien ainsi que beaucoup de moments intenses.

Merci à mon directeur de recherche de m'avoir permis d'en apprendre et d'en comprendre autant. Merci de m'avoir donné l'opportunité de travailler sur des projets passionnants. Merci de m'avoir permis de voyager à travers le monde, ce qui m'a donné l'occasion de rencontrer des scientifiques de renom dont certains sont devenus des collaborateurs.

Merci à tous les professeurs que j'ai côtoyés durant mon parcours universitaire pour leur dévouement à nous transmettre leur passion et leur savoir. Je les remercie aussi pour leur patience et leur indulgence durant mes trop nombreux épisodes d'errance.

Merci au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), à mon directeur de recherche et à mes parents pour leur soutien financier.

Table des matières

Liste des tableaux

Table des figures

Introduction

Au cours des deux dernières décennies, l'informatique quantique a connu un essor remarquable passant d'un domaine émergent quasi ésotérique pour devenir aujourd'hui une discipline à part entière chevauchant informatique, mathématique et physique. Non seulement ce domaine porte l'espoir, voire la promesse, d'un nouveau paradigme pour nos machines de calcul, mais encore il a déjà contribué grandement à l'avancement de ces trois disciplines, que ce soit en théorie de la complexité, en théorie de l'information ou encore en physique de la matière condensée. Un pan important de la recherche dans ce domaine tente de faire le pont entre théorie et pratique. D'un côté, plusieurs travaillent à concevoir en laboratoire un qubit physique alors que de l'autre, plusieurs travaillent à élaborer l'architecture dans laquelle il devrait s'imbriquer.

Supposons donc un instant que nous ayons sous la main un qubit physique, par ex. un qubit supraconducteur tel le transmon [2]. Dans le but de l'opérer, ce qubit est placé dans une cavité électromagnétique. Celle-ci remplit trois rôles importants. Premièrement, elle agit comme filtre sur le bruit de l'environnement et protège ainsi le qubit. Deuxièmement, en récoltant les photons qui s'échappent de la cavité, elle permet d'en mesurer l'état. Troisièmement, en envoyant des impulsions électromagnétiques dans la cavité, elle permet d'appliquer des portes logiques au qubit. Concentrons-nous sur ce dernier point. Pour appliquer une porte logique au qubit, nous envoyons une impulsion de puissance Ω dans la cavité, impulsion qui vient modifier le hamiltonien du système. La porte résultante est une rotation de l'état du qubit autour de l'axe X de la sphère de Bloch [1] :

$$R_X(t) = \mathrm{e}^{-\mathrm{i}\Omega t X}. \tag{1}$$

Un avantage important d'une telle méthode saute aux yeux : l'angle de la rotation dépend du temps d'exposition à l'impulsion. Ceci permet donc très simplement d'appliquer tout un continuum de portes différentes en variant ce temps d'exposition. Malheureusement, ceci

1. Nous notons les matrices de Pauli X, Y, Z plutôt que $\sigma_x, \sigma_y, \sigma_z$.

veut aussi dire que les imprécisions sur le temps d'exposition à l'impulsion ou sur son énergie se répercutent directement dans l'angle de rotation. La précision de l'angle de rotation est alors limitée aux diverses précisions expérimentales. Un autre problème, systématique cette fois, provient de certaines approximations qui ont été faites de prime abord. Par exemple, dans le cas qui nous occupe, un transmon n'est pas un qubit, c'est-à-dire qu'il ne s'agit pas d'un système à deux niveaux. L'effet des niveaux d'énergies négligés, quoique minime, impose une limite à la précision des portes logiques appliquées. Alors, comment pouvons-nous espérer effectuer des calculs nécessitant une plus grande précision que celle imposée par la technologie actuelle ? Doit-on attendre que les méthodes expérimentales se développent davantage ? Heureusement, ce n'est pas le cas. Il existe une solution logicielle (*software*) qui permet de réaliser des composantes effectives plus robustes à partir de celles disponibles. C'est ce qu'on appelle le calcul tolérant aux fautes.

Très simplement, le calcul tolérant aux fautes permet d'obtenir des composantes plus robustes en payant le pris de la redondance. Autrement dit, en affectant davantage de ressources que minimalement nécessaire pour effecter un calcul, on peut en augmenter la précision. Le théorème du *seuil de tolérance aux fautes* [3] formalise cette idée : en supposant un taux d'erreur p_i indépendant pour chacune des composantes c_i d'un circuit, il est possible d'effecteur un calcul de précision arbitraire ε et ce, avec une surcharge (*overhead*) de l'ordre de $\mathcal{O}\big(\text{polylog}(1/\varepsilon)\big)$, à condition que le taux d'erreurs par composante soit plus petit qu'un taux seuil, c.-à-d. $p_i < p_{\text{seuil}}$ pour tout i. Plus précisément, nous encodons un nombre k de qubits logiques dans un nombre n de qubits physiques, où $n > k$. Une conséquence de cet encodage est que nous perdons la propriété de l'Eq. (1) où il était possible d'appliquer facilement un continuum de portes logiques. En fait, bien souvent, les portes qui sont applicables à un qubit encodé de manière tolérante aux fautes forment un petit sous-groupe des portes applicables en principe, c'est-à-dire un sous-groupe des transformations unitaires. Nous devons donc élaborer de nouvelles méthodes pour contourner ce problème. Une de ces solutions est l'injection d'états magiques ou encore la déformation de codes. Nous reviendrons sur ces idées dans les chapitres à venir.

Mes travaux des dernières années ont touché deux aspects importants du calcul tolérant aux fautes. Premièrement, la recherche de codes physiquement réalistes a mené la communauté à s'intéresser aux codes topologiques stabilisateurs. En effet, ceux-ci s'expriment comme des sous-espaces fondamentaux de réseaux de spin 1/2 (qubits) aux hamiltoniens dont les interactions sont à courtes portées et de type Pauli. De plus, un sous-groupe des transformations unitaires, le groupe de Clifford, s'y réalise efficacement. Je me suis intéressé dans un premier temps à classifier les codes topologiques stabilisateurs. Mes collaborateurs et moi avons montré qu'il existe un archétype de code topologique, le code topologique

de Kitaev (CTK) [4]. Tout autre code topologique stabilisateur invariant sous translation peut se ramener à un nombre fini de copies du CTK. Nous avons explicitement construit la transformation qui permet cette classification. Celle-ci s'exprime comme une permutation des opérateurs de Pauli (automorphisme) et s'appliquerait en pratique à l'aide d'un circuit de Clifford local et de profondeur finie. En termes plus physique, c'est dire que les hamiltoniens dont les états fondamentaux correspondent aux codes topologiques stabilisateurs appartiennent tous à la même phase topologique. Je me suis intéressé dans un deuxième temps à généraliser de diverses façons la méthode de décodage par groupe de renormalisation que j'ai développée au cours de ma maîtrise. Tout d'abord, j'ai utilisé la transformation énoncée ci-haut pour décoder d'autres codes topologiques stabilisateurs. Puis, j'ai généralisé la méthode au cas tolérant aux fautes, c'est-à-dire au cas où les mesures de syndromes sont imparfaites. Finalement, je l'ai aussi appliquée au cas où on généralise le CTK en remplaçant les qubits par des qudits, c'est-à-dire des systèmes à d niveaux. Nous reviendrons en plus de détail sur chacun de ces points.

Deuxièmement, comme l'énonce le théorème du seuil de tolérance aux fautes, nous devons payer une « petite » surcharge de $\mathcal{O}\big(\mathrm{polylog}(1/\varepsilon)\big)$ pour effectuer un calcul robuste. Or, en pratique, les constantes cachées dans cette estimation peuvent être importantes, de l'ordre du million, voire du milliard [5]. Cela représente un sérieux obstacle aux espoirs de concevoir un jour un ordinateur quantique. En effet, si chaque qubit logique nécessite un encodage en un million de qubits physiques et que toute opération de base sur celui-ci requiert un million de portes physiques, il est peu probable que nous construisions jamais un ordinateur quantique rivalisant avec un simulateur classique. Durant la deuxième moitié de mon doctorat, j'ai travaillé à réduire la surcharge accompagnant inévitablement le calcul tolérant aux fautes. Pour ce faire, je me suis penché plus spécifiquement sur la notion d'états ressources. En effet, un truc standard pour effectuer une porte quelconque consiste à utiliser une classe réduite de circuits, appelés circuits de Clifford, et un état particulier, appelé état magique. À l'aide de ces deux éléments, on peut étendre la famille de circuits effectifs qu'il est possible de réaliser. La raison pour laquelle on se restreint à l'usage des circuits de Clifford est que pour beaucoup de codes, on sait appliquer ce sous-groupe de circuits de manière non seulement tolérante aux fautes, mais aussi efficace. Dans le but d'obtenir ces états magiques avec grande précision, on fait appel à une procédure nommée « distillation ». La surcharge discutée ci-haut est dominée par cette étape de distillation [6]. La deuxième partie de mes travaux a consisté à étudier de nouvelles familles d'états magiques ainsi qu'à proposer de nouveaux circuits de distillation. À l'aide de ceux-ci, j'ai pu montrer que des gains importants, de l'ordre de 100 ou 1000, étaient possibles par rapport aux standards du moment. En parallèle à mes recherches, plusieurs autres méthodes aux gains comparables ont été développées par d'autres groupes de recherche [7, 8, 9]. Une

autre approche fournissant un ensemble universel de portes consiste à « déformer » le code (passer d'un encodage à un autre) de manière tolérante aux fautes. La stratégie est alors de choisir les différents codes de telle sorte que l'ensemble des portes applicables à ceux-ci soit universel malgré que, pris séparément, les différents codes ne soient pas universels. Mes collaborateurs et moi avons élaboré une stratégie utilisant 15 qubits et deux codes seulement. À ma connaissance, il s'agit de la procédure de déformation la plus efficace connue au moment d'écrire ces lignes.

Cette thèse par articles est constituée de deux parties. La première présente les articles traitant de l'élaboration d'un ensemble universel de portes et la seconde présente ceux en lien avec le décodage de codes topologiques stabilisateurs. Dans chaque partie, les chapitres sont dédiés aux différents articles, à l'exception du premier chapitre de la première partie qui introduit plutôt les états magiques et leur distillation. Chacun des autres chapitres débute par une brève mise en contexte et la description de ma contribution. Suit une introduction à la théorie nécessaire à la compréhension des travaux présentés ou encore des informations supplémentaires étoffant certains passages des articles. L'article publié est enfin inclus tel quel.

Dans l'ensemble de cette thèse, nous supposons que le lecteur est familier avec les notions de base de l'informatique quantique : qubit, sphère de Bloch, circuit quantique, etc. Nous supposons aussi une connaissance minimale des codes stabilisateurs et plus spécifiquement d'une instance en particulier : le code topologique de Kitaev. Autrement, nous renvoyons le lecteur à l'excellent livre d'introduction de Nielsen et Chuang [10] ou encore aux notes de cours de John Preskill [11]. Pour ce qui est des codes stabilisateurs, la thèse de Gottesman [12] est la meilleure introduction à notre connaissance. Le code topologique de Kitaev est décrit en détails dans le mémoire de maîtrise de l'auteur [13].

Finalement, introduisons une convention d'écriture. Au début des différents chapitres traitant des articles se trouve une section « Résumé ». Dans ce résumé, nous référons aux sections des articles en utilisant des chiffres romains, par opposition aux autres sections du chapitre concerné, qui sont notées en chiffres arabes. De plus, nous référons aux figures de l'article en utilisant l'expression « figure », alors que celles du chapitre sont nommées « Fig. ». De manière similaire nous parlons des « tableaux » de l'article et des « Tab. » du chapitre en cours.

Première partie

Ensemble universel de portes

Chapitre 1

Introduction à la distillation d'états magiques

Dans ce chapitre, nous introduisons et étudions différents concepts constituant la base de la distillation d'états magiques : Que sont les états magiques ? En quoi consiste l'injection d'états ? Comment peut-on préparer ces états par la distillation ?

Nous avons vu en introduction la nécessité d'utiliser les codes correcteurs quantiques dans le but de faire du calcul quantique robuste. La classe des codes stabilisateurs [12] joue un rôle très important à cet effet, car elle est relativement simple à étudier tout en permettant le calcul quantique tolérant aux fautes. Nous supposons une certaine familiarité du lecteur avec cette classe, mais nous révisons dans les sections suivantes quelques concepts importants dont le reste de la discussion dépend.

1.1 Portes de Clifford

Les portes de Clifford jouent un rôle particulier dans la théorie des codes stabilisateurs, car tous les circuits d'encodage et de décodage sont en fait des opérations de Clifford. Elles forment un sous-groupe des portes unitaires. Formellement, il s'agit du groupe des permutations (automorphismes) du groupe de Pauli sous conjugaison. En d'autres mots, il s'agit des portes qui transforment un opérateur de Pauli en un autre opérateur de Pauli, tout en préservant l'ensemble des relations de commutations entre ceux-ci.

Des exemples éloquents de portes de Clifford sont donnés par les générateurs de ce groupe. Pour un qubit, le groupe est généré par la porte de Hadamard H, et par la porte

phase S, cf. Eq. (1.1).

$$H = \frac{-1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \qquad S = \begin{pmatrix} e^{-i\pi/4} & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{1.1}$$

Comme une phase globale ne change rien aux transformations, nous avons défini les opérateurs de telle sorte qu'ils appartiennent à SU(2) (unitaires de déterminant 1). On vérifie directement que $HZH = X$ ou encore que $SXS^\dagger = Y$, etc. Notons que le groupe de Pauli est un sous-groupe du groupe Clifford. Le groupe de Clifford à un qubit s'interprète simplement de manière géométrique. En effet, nous pouvons associer les trois éléments non-triviaux du groupe de Pauli aux axes de la sphère de Bloch. Par exemple, on associe l'opérateur Z à l'axe formé par les états propres $|0\rangle$ et $|1\rangle$ de Z. On peut faire de même pour X et Y. L'ensemble des états mixtes qui peuvent être préparés à partir de ceux-ci par combinaison convexe forme un octaèdre dans la sphère de Bloch. Le groupe de Clifford est alors l'ensemble des symétries de cet octaèdre. Par exemple, H échange X et Z et envoie Y vers $-Y$. L'octaèdre est donc préservé.

Pour générer des portes de Clifford sur un nombre arbitraire de qubits, il suffit d'ajouter la porte contrôle-non aux générateurs précédents. Nous notons cette porte $\Lambda_i(X_j)$ où le qubit i est le qubit de contrôle et le qubit j, la cible. Le groupe de Clifford sur n qubits est donc généré par

$$\mathsf{C}_n = \langle H_i, S_i, \Lambda_i(X_j) \rangle. \tag{1.2}$$

Le sous-groupe des portes de Clifford n'est pas dense dans $\mathrm{SU}(2^n)$, c.-à-d. en général une opération quelconque ne peut pas être approximée par une porte de Clifford. En fait, ce sous-groupe est fini. Toutefois, il est maximal, en ce sens que l'ajout de n'importe quelle porte à un qubit n'y appartenant pas déjà suffit à le rendre approximativement universel [14]. Nous reviendrons sur ce point à la section 1.3.

Par abus de langage, nous appelons porte de Clifford tout élément du groupe de Clifford, toute préparation d'un état stabilisateur et toute mesure d'un opérateur de Pauli.

1.2 Portes transverses

Les portes logiques transverses jouent un rôle important dans plusieurs protocoles de calcul tolérant aux fautes, car elles le sont par définition. Étant donné un code à n qubits, une porte logique est transverse si elle s'écrit comme un produit tensoriel de n portes à un qubit :

$$\overline{U} = V^{\otimes n}. \tag{1.3}$$

Ici, U et V représentent deux unitaires quelconques et nous notons $\overline{U}$ la porte U encodée. Cette définition n'est pas la plus générale, car les V pourraient être différents d'un qubit à l'autre, mais la définition Eq. (1.3) nous suffit pour ce qui suit.

Un exemple simple est donné par les portes de Pauli logiques qui sont toujours transverses pour un code stabilisateur. En effet, notons d'abord que par définition, les portes de Pauli sont toujours des produits tensoriels de portes de Pauli à un qubit ; elles sont donc transverses. Ensuite, nous savons que les codes stabilisateurs sont obtenus par une porte d'encodage qui appartient au groupe de Clifford. Par définition du groupe de Clifford, les portes de Pauli encodées seront toujours elle-mêmes des portes de Pauli ; elles sont donc aussi transverses. Un autre exemple est fourni par certaines portes de Clifford. Un code stabilisateur CSS (Calderbank-Shor-Steane) est un code où chaque générateur du stabilisateur n'est composé que de X et $\mathbb{1}$ ou que de Z et $\mathbb{1}$. On peut montrer que pour les codes de cette classe, la porte $\Lambda_i(X_j)$ est transverse. Les codes topologiques de couleurs à bordure triangulaire [15] offrent un exemple particulièrement intéressant où toutes les portes de Clifford sont transverses. Cette propriété est pertinente pour la distillation d'états magiques comme nous le verrons à la section 1.4.

Supposons maintenant qu'un code à n qubits ait une porte transverse $\overline{U} = V^{\otimes n}$. Si ce code a une distance d, il peut alors tolérer jusqu'à $\lfloor d/2 \rfloor$ erreurs. Pour appliquer la porte $\overline{U}$ sur le qubit logique, la porte V est appliquée sur chacun des qubits physiques. Supposons que chaque porte ait une probabilité p d'échouer, indépendamment des autres. Le processus de correction d'erreurs corrige toute erreur sur $\lfloor d/2 \rfloor$ portes ou moins. L'erreur résiduelle est donc de l'ordre de $\mathcal{O}(p^{d/2})$. La porte transverse combinée à la correction d'erreur réduit la probabilité d'erreur des portes physiques $\mathcal{O}(p)$ en probabilité d'erreur logique $\mathcal{O}(p^{d/2})$. De plus, elle ne propage pas les erreurs corrigibles d'un qubit vers d'autres. Elle est donc tolérante aux fautes.

Par contre, une porte qui n'est pas transverse, c'est-à-dire une porte qui nécessite une interaction entre au moins deux qubits du code, risque de propager les erreurs à l'intérieur

du code. Ainsi, une erreur affectant un seul qubit initialement peut se propager à plusieurs qubits suite à l'application de cette porte. En conséquence, certaines erreurs qui étaient corrigibles pourraient ne plus l'être. Dans ce cas, la correction d'erreurs ne parviendrait pas à réduire la probabilité d'erreur logique.

1.3 Injection d'états

Nous avons vu que les portes de Clifford forment un sous-groupe des portes unitaires qu'il est possible d'appliquer sur des qubits. Malheureusement, celui-ci n'est pas dense, c'est-à-dire qu'il existe des portes qui ne peuvent pas être approximées par une porte de Clifford. Or, comme discuté à la section précédente, ces dernières forment malgré tout une classe intéressante dans le cadre des codes stabilisateurs, car elles sont souvent faciles à réaliser, voire transverses. L'injection d'états est une façon de surmonter cette limite. En effet, elle consiste à n'utiliser que des portes de Clifford dans le but d'appliquer une porte qui ne fait pas partie de ce groupe. Cette contradiction apparente vient du fait que l'injection utilise un état particulier appelé « état ressource » ou encore « état magique » qui n'est en général pas un état stabilisateur. Comme l'injection d'états nécessite un circuit de Clifford, il est facile de la rendre tolérante aux fautes en utilisant des codes stabilisateurs. Par contre, la difficulté n'est que repoussée dans la préparation de l'état ressource de grande précision. La solution est fournie par la distillation d'états magiques.

Supposons que nous ayons accès à un état ressource de la forme

$$|Y_\theta\rangle = \frac{1}{\sqrt{2}}\left(\cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle\right) \tag{1.4}$$

dans le plan XZ de la sphère de Bloch. Supposons aussi qu'en ayant choisi judicieusement un code stabilisateur, nous soyons en mesure de faire des portes de Clifford tolérantes aux fautes. L'injection d'états se fait alors à l'aide d'une version encodée du circuit présenté à la Fig. 1.1. Le détail du calcul de son effet sur un état quelconque $|\psi\rangle = a|0\rangle + b|1\rangle$ est donné à l'Eq. (1.5).

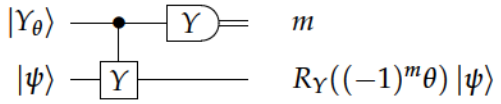


FIGURE 1.1 Circuit d'injection. Nous observons le résultat de mesure $m = 0$ ou $m = 1$ correspondant aux états mésurés $|i\rangle$ ou $|-i\rangle$, respectivement.

$$
\begin{aligned}
|Y_\theta\rangle\,|\psi\rangle =& a\cos\frac{\theta}{2}\,|00\rangle + b\cos\frac{\theta}{2}\,|01\rangle + a\sin\frac{\theta}{2}\,|10\rangle + b\sin\frac{\theta}{2}\,|11\rangle \\
\xrightarrow{\Lambda_1(Y_2)} & a\cos\frac{\theta}{2}\,|00\rangle + b\cos\frac{\theta}{2}\,|01\rangle + ia\sin\frac{\theta}{2}\,|11\rangle + (-i)b\sin\frac{\theta}{2}\,|10\rangle \\
\propto & a\cos\frac{\theta}{2}(|i\rangle + |-i\rangle)\,|0\rangle + b\cos\frac{\theta}{2}(|i\rangle + |-i\rangle)\,|1\rangle \\
& + a\sin\frac{\theta}{2}(|i\rangle - |-i\rangle)\,|1\rangle - b\sin\frac{\theta}{2}(|i\rangle - |-i\rangle)\,|0\rangle \\
= & |i\rangle\left[(a\cos\frac{\theta}{2} - b\sin\frac{\theta}{2})\,|0\rangle + (a\sin\frac{\theta}{2} + b\cos\frac{\theta}{2})\,|1\rangle\right] \\
& + |-i\rangle\left[(a\cos\frac{\theta}{2} + b\sin\frac{\theta}{2})\,|0\rangle + (-a\sin\frac{\theta}{2} + b\cos\frac{\theta}{2})\,|1\rangle\right],
\end{aligned}
\tag{1.5}
$$

où on a utilisé $|0\rangle = \frac{1}{\sqrt{2}}(|i\rangle + |-i\rangle)$, $|1\rangle = \frac{-i}{\sqrt{2}}(|i\rangle - |-i\rangle)$. Lors de la mesure dans la base propre de Y, un résultat $m \in \{0,1\}$ est observé correspondant aux états $|i\rangle$ et $|-i\rangle$, respectivement. L'état résultant est donné à l'Eq. (1.6).

$$
\begin{aligned}
\xrightarrow{m=0} & |i\rangle\left[(a\cos\frac{\theta}{2} - b\sin\frac{\theta}{2})\,|0\rangle + (a\sin\frac{\theta}{2} + b\cos\frac{\theta}{2})\,|1\rangle\right] \\
= & |i\rangle\,(\cos\frac{\theta}{2}\mathbb{1} + \sin\frac{\theta}{2}XZ)\,|\psi\rangle \\
= & |i\rangle\exp(-i\frac{\theta}{2}Y)\,|\psi\rangle = |i\rangle\,R_Y(\theta)\,|\psi\rangle \\
\xrightarrow{m=1} & |-i\rangle\left[(a\cos\frac{\theta}{2} + b\sin\frac{\theta}{2})\,|0\rangle + (-a\sin\frac{\theta}{2} + b\cos\frac{\theta}{2})\,|1\rangle\right] \\
= & |-i\rangle\,(\cos\frac{\theta}{2}\mathbb{1} - \sin\frac{\theta}{2})XZ\,|\psi\rangle \\
= & |-i\rangle\exp(i\frac{\theta}{2}Y)\,|\psi\rangle = |i\rangle\,R_Y(-\theta)\,|\psi\rangle
\end{aligned}
\tag{1.6}
$$

L'injection de l'état $|Y_\theta\rangle$ applique la rotation $R_Y(\theta)$ (cas $m = 0$) ou $R_Y(-\theta) = R_Y^\dagger(\theta)$ (cas $m = 1$) aléatoirement, dépendamment du résultat de la mesure. Le fait que l'angle de rotation soit aléatoire représente une complication supplémentaire, mais celle-ci n'est pas insurmontable, car l'angle de rotation appliqué est tout de même connu.

Dans le but d'utiliser l'injection d'états pour compléter notre groupe de portes approxi-

mativement universel, il suffit qu'une source d'états non-stabilisateurs soit disponible. De plus, pour qu'un état $|Y_\theta\rangle$ soit non-stabilisateur il suffit que $\theta \neq m\frac{\pi}{2}$, où m est un entier quelconque. L'état $|Y_{\pi/4}\rangle$ est typiquement utilisé et on le nomme souvent $|H\rangle$, car il s'agit de l'état propre +1 de la transformation de Hadamard.

1.4 Distillation d'états magiques

Comme nous l'avons vu aux sections précédentes, nous sommes restreints à la préparation d'états stabilisateurs et à l'application de portes de Clifford. Or, pour compléter notre ensemble universel de portes, nous avons aussi vu que l'on pouvait utiliser une source d'états non-stabilisateurs pour injecter des portes non-Clifford à l'aide d'un circuit de Clifford. Le problème de la préparation de ces états ressources demeure. La distillation offre une solution.

Comme l'injection, la distillation d'états ne fait intervenir que des circuits de Clifford et suppose une source d'états non-stabilisateurs. Par contre, ces états ne sont pas supposés « parfaits ». Nous supposons plutôt qu'il s'agit d'états mixtes suffisamment près de l'état ressource idéal. La distillation permet alors de prendre plusieurs copies imparfaites de l'état magique et d'en retirer un nombre moindre, mais de plus grande qualité. Plus formellement, supposons que notre source fournisse des états ρ pour lesquels $D(\rho, |Y_\theta\rangle\langle Y_\theta|) = \epsilon$, où D est une mesure de distance entre opérateurs, par ex. la norme de trace. Un protocole de distillation permet de prendre n copies de ρ et d'obtenir $m < n$ copies d'un nouvel état ρ' tel que $D(\rho', |Y_\theta\rangle\langle Y_\theta|) \sim \alpha\epsilon^\beta$ où $\alpha \lesssim \binom{n}{\beta}$ est un facteur combinatoire et $\beta \geq 1$. Dans les discussions à venir, nous dirons qu'un état magique imparfait est plus « propre » qu'un autre s'il est plus près de l'état magique idéal.

Plus concrètement, nous nous attardons pour la suite de ce chapitre à deux exemples simples. Il s'agit du premier protocole de distillation introduit par Bravyi et Kitaev [16] et d'un autre introduit par Reichardt [17]. [1] Avant d'étudier ces exemples concrets, nous introduisons brièvement des concepts importants.

1. Il n'est pas nécessaire de lire ces articles pour la suite.

1.4.1 Seuil, erreur résiduelle et rendement

L'erreur sur un état magique approximatif est souvent calculée à l'aide d'une mesure de distance, la distance de trace par exemple. Nous notons ϵ cette distance. À la suite du protocole de distillation l'erreur résiduelle a la forme $\epsilon_{\text{out}} = \alpha\epsilon_{\text{in}}^{\beta} + \mathcal{O}(\epsilon^{\beta+1})$. L'erreur seuil, ϵ_{seuil}, à partir de laquelle la qualité de l'état est améliorée par la distillation est donnée par $\epsilon_{\text{out}} = \epsilon_{\text{in}}$. Au-delà de cette erreur seuil, l'état est trop dissemblable de l'état ressource désiré et il n'est pas possible de le distiller à l'aide du protocole considéré.

Certains protocoles de distillation ont également une probabilité de rejet, que nous notons δ ici. La probabilité d'accepter l'état à la sortie est donc $1 - \delta$. De plus, le protocole consomme n copies de l'état magique imparfait pour en redonner m de plus grande précision. On appelle le rendement du protocole le ratio $\gamma = \frac{m}{n}(1 - \delta)$ qui représente le nombre moyen d'états améliorés attendus par état investi. En appliquant le protocole de manière itérative k fois nous obtenons γ^k états distillés par état investi avec une erreur résiduelle de l'ordre de ϵ^{β^k}. Le nombre de copies imparfaites consommées est exponentiel alors que l'erreur est supprimée de manière doublement exponentielle, ce qui rend le protocole efficace.

1.4.2 Distillation avec le code à cinq qubits

Bravyi et Kitaev [16] ont proposé d'utiliser le code à cinq qubits [18, 19], dans le but de distiller l'état $|A\rangle = \cos\beta\,|0\rangle + e^{i\pi/4}\cos\beta\,|1\rangle$ où $\cos 2\beta = 1/\sqrt{3}$.[2] Il est plus facile de visualiser $|A\rangle$ en l'écrivant sous la forme d'une matrice densité $\rho_A = \frac{1}{2}(\mathbb{1} + \frac{1}{\sqrt{3}}(X + Y + Z))$. Il s'agit de l'état sur la sphère de Bloch dans la direction $(1, 1, 1)$.

Le code à cinq qubits est un code stabilisateur encodant un qubit logique. Le groupe stabilisateur est donc généré par quatre opérateurs. Un exemple de générateur et d'opérateurs logiques est donné au Tab. 1.1. Cet ensemble de générateurs est particulièrement simple puisqu'il s'agit des permutations cycliques de $XZZX\mathbb{1}$. D'ailleurs, notons que $s_1s_2s_3s_4 = ZZX\mathbb{1}X$ est aussi une de ces permutations et pourrait remplacer n'importe lequel des quatre générateurs introduits. Nous notons le groupe stabilisateur correspondant $\mathcal{S}$ et le code $\mathcal{C}_{\mathcal{S}}$. Ce code possède une porte de Clifford transverse, $A = SH$. Dans le but de mieux comprendre son effet, explicitons comment les matrices de Pauli sont transformées par cette

2. Dans la publication, cet état est appelé $|T\rangle$, mais aujourd'hui, T fait plutôt référence à la rotation d'angle $\pi/4$ autour de l'axe Z de la sphère de Bloch et à l'état correspondant $|T\rangle = |0\rangle + e^{i\pi/4}\,|1\rangle$.

$$
\begin{aligned}
s_1 &= XZZX\mathbb{1} \\
s_2 &= \mathbb{1}XZZX \\
s_3 &= X\mathbb{1}XZZ \\
s_4 &= ZX\mathbb{1}XZ \\
\hline\hline
\overline{X} &= XXXXX \\
\overline{Z} &= ZZZZZ
\end{aligned}
$$

Tableau 1.1 Générateurs du stabilisateur et opérateurs logiques du code à cinq qubits.

opération, cf. Eqs. 1.7-1.9.

$$AXA^\dagger = Z \tag{1.7}$$
$$AYA^\dagger = X \tag{1.8}$$
$$AZA^\dagger = Y \tag{1.9}$$

L'opérateur de Clifford A permute de manière cyclique les matrices de Pauli. Sans même faire la diagonalisation explicitement, nous déduisons de ce point de vue géométrique que A est une rotation d'angle $2\pi/3$ autour de l'axe $(1,1,1)$. Par conséquent, $|A\rangle$ est un état propre de A. En effet, comme l'axe de rotation coupe la sphère de Bloch au point correspondant à $|A\rangle$, une rotation autour de cet axe ne peut modifier cet état, c.-à-d. elle ne peut que lui ajouter une phase. Ceci n'est pas un coïncidence comme nous le verrons. De plus, comme cette rotation de SU(2) est d'ordre trois, ses valeurs propres doivent être $e^{\pm i2\pi/3}$.

Pour vérifier que A est une porte transverse de $\mathcal{C}_\mathcal{S}$, il faut vérifier que $\mathcal{S}$ est invariant sous l'application de $A^{\otimes 5}$, c.-à-d. $A^{\otimes 5}\mathcal{S}(A^{\otimes 5})^\dagger = \mathcal{S}$. Notons, qu'il n'est pas nécessaire que chaque élément soit individuellement invariant. Pour alléger la notation, nous écrivons A_5 pour signifier $A^{\otimes 5}$. Nous vérifions

$$A_5 s_1 A_5^\dagger = ZYYZ\mathbb{1} = s_1 s_3 s_4, \tag{1.10}$$
$$A_5 s_2 A_5^\dagger = \mathbb{1}ZYYZ = s_1 s_2 s_4, \tag{1.11}$$
$$A_5 s_3 A_5^\dagger = Z\mathbb{1}ZYY = s_1 s_2 s_3, \tag{1.12}$$
$$A_5 s_4 A_5^\dagger = YZ\mathbb{1}ZY = s_2 s_3 s_4. \tag{1.13}$$

En inversant les Eqs. 1.10-1.13, nous obtenons les anciens générateurs en fonction des nouveaux. Nous en concluons que le groupe dans son ensemble est préservé. Ceci montre que l'opération transverse $A^{\otimes 5}$ est une opération logique, c.-à-d. elle préserve le code $\mathcal{C}_\mathcal{S}$.

Pour découvrir son effet logique, il faut étudier comment les opérateurs de Pauli logiques se transforment sous son action. À l'aide des Eqs. 1.7-1.9, nous avons

$$A_5 \overline{X} A_5^\dagger = \overline{Z}, \tag{1.14}$$
$$A_5 \overline{Y} A_5^\dagger = \overline{X}, \tag{1.15}$$
$$A_5 \overline{Z} A_5^\dagger = \overline{Y}. \tag{1.16}$$

Nous en concluons que l'opérateur A appliqué de manière transverse est une opération logique sur le code et qu'il s'agit précisément de la transformation A, c.-à-d. $A_5 = \overline{A}$.

Considérons maintenant la distillation de l'état $|A\rangle$. De manière similaire à ce qui a été fait plus haut, nous posons $|A_i\rangle = |A\rangle^{\otimes i}$. L'état à cinq qubits $|A_5\rangle$, étant un état produit, ne peut être un état du code à cinq qubits, $|A_5\rangle \notin \mathcal{C}_\mathcal{S}$. Par contre, c'est un état propre de $\overline{A}$, car chaque $|A\rangle$ est individuellement un état propre de A. Comme $A|A\rangle = e^{i2\pi/3}|A\rangle$, nous avons

$$\overline{A}|A_5\rangle = A_5|A_5\rangle = e^{i10\pi/3}|A_5\rangle = e^{-i2\pi/3}|A_5\rangle. \tag{1.17}$$

Ce protocole de distillation consiste à mesurer les générateurs du groupe stabilisateur dans le but de projeter $|A_5\rangle$ sur le code $\mathcal{C}_\mathcal{S}$. Le résultat de la mesure est probabiliste, mais nous ignorons ce détail pour l'instant et nous supposons que nous pouvons projeter $|A_5\rangle$ sur le code $\mathcal{C}_\mathcal{S}$. L'opérateur de projection s'écrit $\prod_\mathcal{S} = \frac{1}{16}\prod_{i=1}^4(\mathbb{1} + s_i)$. Or, comme $\overline{A}$ est une opération logique, celle-ci commute avec $\prod_\mathcal{S}$. Donc, $\prod_\mathcal{S}|A_5\rangle$ est forcément un état propre de l'opérateur logique $\overline{A}$, c.-à-d.

$$\overline{A}\prod_\mathcal{S}|A_5\rangle = \prod_\mathcal{S}\overline{A}|A_5\rangle = e^{-i2\pi/3}\prod_\mathcal{S}|A_5\rangle. \tag{1.18}$$

De plus, il s'agit de l'état propre à valeur propre $e^{-i2\pi/3}$, c.-à-d. $\prod_\mathcal{S}|A_5\rangle$ est l'état $|A^\perp\rangle$ encodé. Considérons maintenant l'état $|A_4\rangle \otimes |A^\perp\rangle$. Nous avons alors $A_5|A_4\rangle \otimes |A^\perp\rangle = e^{i6\pi/3}|A_4\rangle \otimes |A^\perp\rangle = |A_4\rangle \otimes |A^\perp\rangle$ et donc,

$$\overline{A}\prod_\mathcal{S}|A_4\rangle \otimes |A^\perp\rangle = \prod_\mathcal{S} A_5|A_4\rangle \otimes |A^\perp\rangle = \prod_\mathcal{S}|A_4\rangle \otimes |A^\perp\rangle. \tag{1.19}$$

Nous voyons que $\prod_\mathcal{S}|A_4\rangle \otimes |A^\perp\rangle$ est un état propre de $\overline{A}$ à valeur propre +1. Or, +1 n'est pas une valeur propre de $\overline{A}$. Nous en déduisons que $\prod_\mathcal{S}|A_4\rangle \otimes |A^\perp\rangle = 0$. Ce raisonnement est valide pour tout état produit tensoriel de quatre copies de $|A\rangle$ et d'une copie de $|A^\perp\rangle$, leur ordre ne jouant aucun rôle ici.

Poursuivons notre analyse et considérons cinq préparations imparfaites de $|A\rangle$. Une préparation réaliste de cet état résulte en une matrice densité qui pourrait à priori se retrouver n'importe où dans la « boule » de Bloch. Toutefois, grâce à la porte A elle-même, nous pouvons appliquer une opération de déphasage dans la base définie par $\{|A\rangle, |A^\perp\rangle\}$:

$$D(\rho) = \frac{1}{3}(\rho + A\rho A^\dagger + A^\dagger \rho A). \tag{1.20}$$

Une façon simple de se convaincre que l'opération D déphase dans la base $|A\rangle$ est d'étudier son action sur les opérateurs de Pauli. Tout opérateur de Pauli non-trivial, $P \in \{X, Y, Z\}$, sur lequel D est appliquée résulte en

$$D(P) = \frac{1}{3}(X + Y + Z), \tag{1.21}$$

ce qui correspond à l'axe attendu. Nous ramenons tout état imparfait approximant $|A\rangle$ à la forme Eq. (1.22) en appliquant cette opération de déphasage.

$$\rho_A = (1-\epsilon)|A\rangle\langle A| + \epsilon|A^\perp\rangle\langle A^\perp|, \tag{1.22}$$

c.-à-d. un état se trouvant le long de l'axe de rotation de A. Au long, l'état à cinq qubits s'écrit

$$\begin{aligned}\rho_A^{\otimes 5} =& (1-\epsilon)^5|A_5\rangle\langle A_5| + (1-\epsilon)^4\epsilon|A_4A^\perp\rangle\langle A_4A^\perp| & (1.23)\\ &+ (1-\epsilon)^4\epsilon|A_3A^\perp A\rangle\langle A_3A^\perp A| + \cdots + (1-\epsilon)^4\epsilon|A^\perp A_4\rangle\langle A^\perp A_4| + \mathcal{O}(\epsilon^2) & (1.24)\end{aligned}$$

En projetant ensuite sur le code, nous éliminons toute contribution des termes dont le préfacteur est au premier ordre en ϵ, c.-à-d.

$$\prod_{\mathcal{S}} \rho_A^{\otimes 5} \prod_{\mathcal{S}} \propto (1-\epsilon)^5 \prod_{\mathcal{S}} |A_5\rangle\langle A_5| \prod_{\mathcal{S}} + \mathcal{O}(\epsilon^2)\tilde{\rho}. \tag{1.25}$$

Une fois décodé, l'état résultant est $\rho_A \to |A^\perp\rangle\langle A^\perp| + \mathcal{O}(\epsilon^2)\tilde{\rho}$, où $\tilde{\rho}$ peut être calculé et où une normalisation s'impose. Pour ϵ plus petit qu'un ϵ_{seuil} déterminé à l'aide des constantes cachées dans la notation $\mathcal{O}$, nous avons à la sortie de la distillation un état plus propre que les états à l'entrée. Notons que nous pouvons passer de $|A^\perp\rangle$ à $|A\rangle$ à l'aide de l'opération de Clifford YH, car $|A^\perp\rangle = YH|A\rangle$. Pour nous en convaincre, remarquons que $YH(X+Y+Z)HY = -(X+Y+Z)$.

En mesurant les générateurs du stabilisateur et en sélectionnant à postériori les instances où le résultat de la mesure est le syndrome trivial, nous projetons sur le sous-espace code

s_1^X	=	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	X	X	X	X
s_2^X	=	$\mathbb{1}$	X	X	$\mathbb{1}$	$\mathbb{1}$	X	X
s_3^X	=	X	$\mathbb{1}$	X	$\mathbb{1}$	X	$\mathbb{1}$	X
s_1^Z	=	$\mathbb{1}$	$\mathbb{1}$	$\mathbb{1}$	Z	Z	Z	Z
s_2^Z	=	$\mathbb{1}$	Z	Z	$\mathbb{1}$	$\mathbb{1}$	Z	Z
s_3^Z	=	Z	$\mathbb{1}$	Z	$\mathbb{1}$	Z	$\mathbb{1}$	Z
$\overline{X}$	=	X	X	X	X	X	X	X
$\overline{Z}$	=	Z	Z	Z	Z	Z	Z	Z

Tableau 1.2 Générateurs du stabilisateur et opérateurs logiques du code à sept qubits.

$\mathcal{C}_\mathcal{S}$. En ce faisant nous éliminons toute contribution des états formés d'une copie de $|A^\perp\rangle$. Pour tout autre résultat, l'état est simplement écarté. Pour connaître la probabilité de succès asymptotique, il suffit de calculer (par ex. à l'aide de Mathematica) la norme de l'état $|A_5\rangle$ une fois projeté sur le code. Nous obtenons

$$\langle A_5| \prod_{\mathcal{S}} |A_5\rangle = \frac{1}{6}. \tag{1.26}$$

La probabilité de succès du protocole est donc asymptotiquement ($\epsilon \to 0$) de $1/6$. Par conséquent, le rendement asymptotique du protocole se résume ainsi : cinq états sont nécessaires pour en produire un plus propre avec probabilité $1/6$. Le rendement est donc de 30 pour 1 pour une erreur résiduelle d'ordre $\mathcal{O}(\epsilon^2)$. Cette probabilité de succès qui sature à une valeur plus petite que un n'est pas souhaitable. Heureusement, ceci n'est pas une caractéristique typique des protocoles de distillation.

1.4.3 Distillation avec le code à sept qubits

Voyons maintenant comment Reichardt [17] s'est servi du code à sept qubits introduit par Steane [20] pour distiller des états $|H\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle$. Cet exemple est important, car l'état $|H\rangle$ et le code à sept qubits apparaissent dans les articles des chapitres à venir. Les générateurs du stabilisateur du code à sept qubits ainsi que ses opérateurs logiques sont présentés au Tab. 1.2. Ce code appartient à la famille des codes CSS pour lesquels il existe un ensemble générateur où chaque opérateur n'est constitué que de X et $\mathbb{1}$ ou bien que de Z et $\mathbb{1}$. Dans ce qui suit, nous distinguons les deux sous-groupes X et Z du stabilisateur. Nous les nommons $\mathcal{S}_X = \langle s_i^X \rangle$ et $\mathcal{S}_Z = \langle s_i^Z \rangle$.

Nous pouvons facilement nous convaincre que la porte transverse $H^{\otimes 7} = H_7$ est l'opération logique H. Tout d'abord rappelons que H « échange » X et Z, c.-à-d. $HZH = X$ et vice-versa. Puis, notons que le support des générateurs est le même pour les opérateurs s_i^X et s_i^Z. Nous en concluons que pour tout i, $Hs_i^X H = s_i^Z$ et vice-versa. Par conséquent, les sous-groupes $\mathcal{S}_X$ et $\mathcal{S}_Z$ ne sont qu'échangés et le groupe $\mathcal{S}$ est préservé dans son ensemble. Donc, H_7 est transverse. Ensuite, notons que le support des opérateurs logiques est aussi le même pour $\overline{X}$ et $\overline{Z}$ et alors $H_7\overline{X}H_7 = \overline{Z}$. H_7 est une opération logique qui « échange » $\overline{X}$ et $\overline{Z}$, c.-à-d. $H_7 = \overline{H}$.

De manière similaire à la distillation du code à cinq qubits, nous avons que $|H_7\rangle$ n'est pas un état du code, mais est tout de même un état propre de $\overline{H}$. Pour effectuer la distillation, nous le projetons sur le code. Dans ce cas-ci, il est possible de calculer explicitement le recouvrement entre le code et les états sans erreurs ou bien avec une seule erreur. Regardons de plus près l'état $|H_7\rangle$ sous forme d'opérateur densité. Nous avons

$$|H_7\rangle\langle H_7| = \frac{1}{2^7}(\mathbb{1} + H)^{\otimes 7} \tag{1.27}$$

$$= \frac{1}{2^7}(\mathbb{1}_7 + \sum_{i=1}^{7} H_i + \sum_{i<j} H_i \otimes H_j + \cdots + H^{\otimes 7}). \tag{1.28}$$

Aussi, remarquons que tous les éléments non-triviaux des stabilisateurs $\mathcal{S}_X$ et $\mathcal{S}_Z$ sont de poids quatre. Le projecteur sur le code s'écrit

$$\Pi_{\mathcal{S}_X} = \frac{1}{2^3} \sum_{s \in \mathcal{S}_X} s, \tag{1.29}$$

$$\Pi_{\mathcal{S}_Z} = \frac{1}{2^3} \sum_{s \in \mathcal{S}_Z} s, \tag{1.30}$$

où pour tout $s \in \mathcal{S}_X - \{\mathbb{1}\}$ ou $s \in \mathcal{S}_Z - \{\mathbb{1}\}$, $|s| = 4$. Puis, notons une série de propriétés importantes. Tout d'abord, remarquons que

$$\mathrm{Tr}\,(H) = \mathrm{Tr}\,(X) = \mathrm{Tr}\,(Z) = \mathrm{Tr}\,(Y) = 0, \tag{1.31}$$

$$\mathrm{Tr}\,(XH) = \mathrm{Tr}\,(ZH) = \sqrt{2}, \tag{1.32}$$

$$\mathrm{Tr}\,(XZH) = 0. \tag{1.33}$$

Rappelons aussi qu'en général

$$\mathrm{Tr}\,(A \otimes B) = \mathrm{Tr}\,(A)\,\mathrm{Tr}\,(B). \tag{1.34}$$

Le recouvrement se calcule comme suit

$$\text{Tr } (\Pi_{\mathcal{S}_X}\Pi_{\mathcal{S}_Z}|H_7\rangle\langle H_7|). \tag{1.35}$$

Pour déduire la valeur de cette trace, nous étudions les différents termes qui se présentent dans la somme et qui résultent du produit des expressions 1.28, 1.29 et 1.30. Tout d'abord, nous pouvons déjà conclure que le terme qui est le produit des opérateurs identité a une contribution $1/64$. Ensuite, dès qu'un terme a une composante XZ ou bien XZH, alors sa trace est nulle, cf. Eq. (1.31), Eq. (1.33) et Eq. (1.34). Nous en concluons que la partie X et la partie Z du stabilisateur ne peuvent avoir de support en commun. Or, comme les éléments non-triviaux des deux stabilisateurs sont tous de poids quatre et qu'il n'y a que sept qubits au total, cela est impossible, sauf si une des deux parties est triviale. En d'autres mots, les termes contribuant à la trace ont soit leur partie X, soit leur partie Z, triviale. De plus, comme H a trace nulle, il faut aussi que le support du stabilisateur et celui du terme en H (Eq. (1.28)) coïncident, ce qui n'est possible que pour certains termes de poids quatre. Étant donné un tel terme, seulement deux stabilisateurs sont possibles, un X et un autre Z. Chacun de ces termes a une contribution $1/256$. Il y a sept termes non-triviaux par type de stabilisateurs, c.-à-d. $|\mathcal{S}_X - \{\mathbb{1}\}| = |\mathcal{S}_Z - \{\mathbb{1}\}| = 7$, donc 14 termes au total. La trace de l'Eq. (1.35) vaut donc $1/64 + 14/256 = 9/128$.

Dans le cas où une erreur s'est produite, nous avons plutôt comme état initial quelque chose comme

$$|H_6H^\perp\rangle\langle H_6H^\perp| = \frac{1}{2^7}(\mathbb{1} + H)^{\otimes 6} \otimes (\mathbb{1} - H). \tag{1.36}$$

Certains termes pertinents à la trace permettant de calculer le recouvrement sont affectés d'un signe moins, dû au dernier facteur de l'Eq. (1.36). En fait, exactement huit des 14 termes sont affectés, car chaque qubit participe à exactement quatre stabilisateurs $\mathcal{S}_X$ et $\mathcal{S}_Z$ non-triviaux, cf. Tab. 1.2. Au total, nous avons que la trace vaut $1/64 + 6/256 - 8/256 = 1/128$. En résumé, nous avons

$$\text{Tr } (\Pi_{\mathcal{S}_X}\Pi_{\mathcal{S}_Z}|H_7\rangle\langle H_7|) = 9/128, \tag{1.37}$$

$$\text{Tr } \left(\Pi_{\mathcal{S}_X}\Pi_{\mathcal{S}_Z}|H_6H^\perp\rangle\langle H_6H^\perp|\right) = 1/128, \tag{1.38}$$

$$\dots$$

$$\text{Tr } \left(\Pi_{\mathcal{S}_X}\Pi_{\mathcal{S}_Z}|H^\perp H_6\rangle\langle H^\perp H_6|\right) = 1/128. \tag{1.39}$$

L'état passe donc d'une erreur ϵ à une erreur résiduelle $7\epsilon/9 + \mathcal{O}(\epsilon^2)$. La probabilité de succès asymptotique ($\epsilon \to 0$) est de $9/128$. Le rendement est donc de 896 pour 9 ($\sim 1\%$).

Ce protocole comporte deux problèmes majeurs le rendant inutile en pratique : l'erreur résiduelle ne décroit qu'exponentiellement, car $\beta = 1$, et le rendement est très faible.

La quête de protocoles plus efficaces que les deux présentés ci-haut a motivé la recherche dans ce domaine. En effet, la distillation semble être le goulot d'étranglement principal du calcul tolérant aux fautes. C'est à ce problème que les articles suivants s'attaquent.

Chapitre 2

Article : Distillation of non-stabilizer states for universal quantum computation

Guillaume Duclos-Cianci, Krysta M. Svore, *Distillation of non-stabilizer states for universal quantum computation*, Phys. Rev. A 88, 042325 (2013).

2.1 Contexte

À l'été 2012, j'ai eu la chance de faire un stage chez Microsoft Research (Redmond, WA) dans le groupe *Quantum Architecture and Computation* (QuArC) sous la supervision de Krysta Svore. L'objectif initial du projet était de programmer les différents protocoles de distillation existants en utilisant la suite logicielle « Liqui$|\rangle$ » (prononcé *liquid*) en cours de développement chez eux. Celle-ci permet de simuler classiquement un ordinateur quantique basé sur le modèle des circuits. Après m'être familiarisé avec la littérature sur le sujet durant les premières semaines, je leur ai plutôt proposé de me laisser explorer quelques idées qui m'étaient venues lors de mon étude. Un point particulier des différents protocoles de distillation proposés jusqu'alors était que les états d'entrée et de sortie de ces protocoles étaient toujours les mêmes. À cela s'ajoutait l'intuition qu'un ensemble de portes élémentaires sur-complet pourrait réduire le coût de compilation de diverses portes. Mon idée était donc de concevoir un protocole de distillation qui modifierait les états en cours de distillation. Je n'y suis que partiellement arrivé. J'ai proposé une famille d'états ressources obtenus de manière itérative à partir d'un état ressource racine qui doit lui-même être distillé à l'aide de

protocoles déjà établis. J'ai montré que l'usage des états appartenant à cette nouvelle famille permet d'effectuer des rotations autour des axes de Pauli avec, en moyenne, un nombre de portes réduit par rapport aux protocoles existant au moment de la publication. Une autre utilité de cette famille d'états est qu'elle permette d'appliquer la porte $V_Y = R_Y(\arccos(\frac{3}{5}))$, qui elle est efficacement dense dans SU(2) [21]. Ces travaux ont aussi mené à une demande de brevet par Mircosoft [22]. Krysta et moi avons tous deux participé à la rédaction de l'article à parts égales. Ces travaux ont été acceptés à la conférence *Theory of Quantum Computation, Communication and Cryptography (TQC) 2013*, mais je n'ai pu me présenter à la conférence pour des raisons de santé.

2.2 Résumé

Les sections I et II introduisent le travail et rappellent l'injection d'états. La section 2.3 ci-dessous introduit les notions de coûts directs et différés qui apparaissent tout au long de l'article. Bâtissant sur les protocoles existants, nous supposons que nous avons à notre disposition une banque d'états $|H\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle$ arbitrairement propres. Comme cela est expliqué à la section III, nous pouvons à l'aide de ces états et d'un circuit de Clifford très simple bâtir de manière récursive la famille d'états $|H_i\rangle = \cos\theta_i|0\rangle + \sin\theta_i|1\rangle$, où $\cot\theta_i = \cot^{i+1}\frac{\pi}{8}$. À l'aide de circuits de Clifford supplémentaires, nous pouvons aussi « densifier » cette famille. La section 2.4 élabore sur les différents trucs que nous avons utilisés pour construire ces circuits. Nous appelons cette famille « échelle d'états » (*ladder of states*). À l'aide des états $|H_i\rangle$, nous pouvons réaliser l'ensemble de portes $R_Y(\pm 2\theta_i)$. Comme l'angle de rotation appliqué est aléatoirement $\pm 2\theta_i$ et que ce signe ne peut être corrigé en général, les protocoles de compilation usuels ne s'appliquent pas. Nous avons plutôt opté pour une approche qui est à la fois directe et dynamique. La compilation est discutée à la section IV. La section V compare différentes versions de notre protocole à d'autres déjà établis au moment de la publication. La section 2.7 présente quelques compléments de résultats qui n'ont pas été inclus dans la publication. Cette dernière section devrait être lue à la suite de l'article.

2.3 Coûts directs et différés

Introduisons la distinction entre coûts directs (*online cost*) et coûts différés (*offline cost*). Une architecture basée sur les états magiques fonctionnerait de la manière suivante. Une partie de l'ordinateur quantique (probablement la majorité) produit sans cesse des états magiques. Ceux-ci sont alors utilisés par le module de calcul qui n'opère que des transformations de Clifford, assistées des états magiques. La somme des états ressources imparfaits impliqués dans la production des états magiques est comptabilisée en tant que coût différé tandis que le nombre d'états magiques distillés utilisés par le calculateur pour appliquer une porte logique est plutôt comptabilisé en tant que coût direct. Un des intérêts de notre protocole est qu'il permet différents compromis (*trade-offs*) entre coûts directs et coûts différés.

2.4 Jongler avec les circuits de Clifford

Le protocole décrit dans l'article nécessite un état non-stabilisateur racine (*seed*), comme le montre la figure 2 de l'article. Le plus simple est de prendre l'état $|H\rangle$ lui-même comme point de départ. Toutefois, ce choix n'est pas unique. Nous préparons aussi de nouveaux états non-stabilisateurs à l'aide de circuits de Clifford. Ceux-ci sont illustrés à la figure 4 et au tableau I de l'article. Nous avons d'abord découvert les stabilisateurs du tableau I, par essais et erreurs, puis nous avons construit les circuits de la figure pour les décoder. Nous avons choisi les stabilisateurs de manière à ce que si les états d'entrée sont dans le plan XZ de la sphère de Bloch, alors l'état projeté sur le code, puis décodé, se trouve lui aussi dans ce plan.

Pour mieux comprendre comment nous avons construit les circuits, nous énumérons à la Fig. 2.1 quelques identités utiles. Ces identités se démontrent directement. Nous donnons

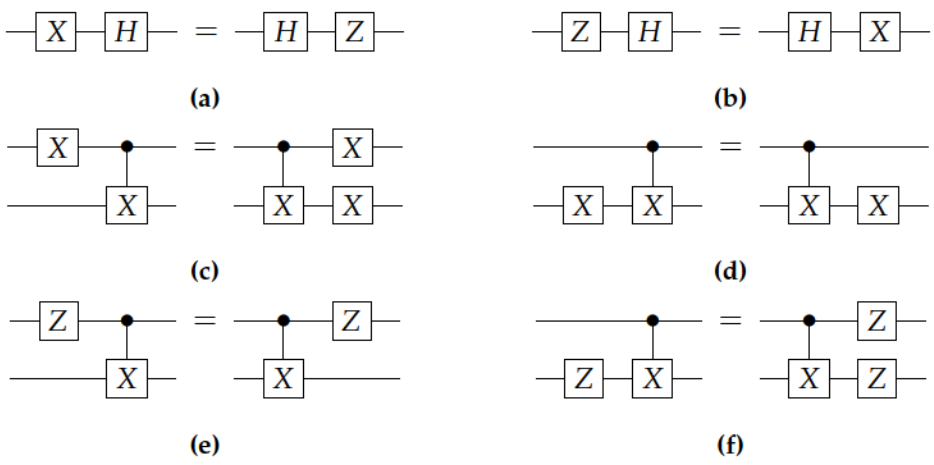


FIGURE 2.1 Identités utiles pour manipuler des circuits de Clifford.

en exemple aux Eqs. 2.1-2.5 le calcul démontrant celle de la Fig. 2.1c.

$$X_1 \cdot \Lambda_1(X_2) = X_1 \cdot \left[\frac{1}{2}(\mathbb{1}_1 + Z_1) \otimes \mathbb{1}_2 + \frac{1}{2}(\mathbb{1} - Z_2) \otimes X_2\right] \tag{2.1}$$

$$= \left[\frac{1}{2}(\mathbb{1}_1 - Z_1) \otimes \mathbb{1}_2 + \frac{1}{2}(\mathbb{1} + Z_1) \otimes X_2\right] \cdot X_1 \tag{2.2}$$

$$= \left[\frac{1}{2}(\mathbb{1}_1 + Z_1) \otimes \mathbb{1}_2 + \frac{1}{2}(\mathbb{1} - Z_2) \otimes X_2\right] \cdot X_2 \cdot X_1 \tag{2.3}$$

$$= \left[\frac{1}{2}(\mathbb{1}_1 + Z_1) \otimes \mathbb{1}_2 + \frac{1}{2}(\mathbb{1} - Z_2) \otimes X_2\right] \cdot X_1 \otimes X_2 \tag{2.4}$$

$$= \Lambda_1(X_2) \cdot X_1 \otimes X_2 \tag{2.5}$$

Pour arriver à bâtir intuitivement des circuits à partir d'un ensemble générateur d'un code stabilisateur, nous notons la propriété suivante. Considérons un circuit de Clifford quelconque $\mathcal{C}$ permettant un encodage des générateurs de Pauli, $\{Z_i, X_j\}$, en $\{s_i, t_j\}$, cf. Eq. (2.6).

$$s_i = \mathcal{C} Z_i \mathcal{C}^\dagger \qquad\qquad t_i = \mathcal{C} X_i \mathcal{C}^\dagger \tag{2.6}$$

Supposons que nous désirions effectuer un changement de générateurs. Par exemple, nous pourrions effectuer un changement de générateur « élémentaire », c.-à-d. le plus simple possible, en remplaçant un générateur par son produit avec un autre : $s_i \to s_i s_j$. Comme les circuits de Clifford préservent la structure du groupe de Pauli, les relations de commutation des différents générateurs doivent rester invariantes. Ceci implique que la substitution d'un stabilisateur s_i doit toujours être appliquée en dualité pour le t_j correspondant, c.-à-d.

$t_j \to t_i t_j$. L'Eq. (2.7) donne la forme générale d'un tel changement élémentaire.

$$s_i \to s_i' = s_i s_j \qquad\qquad t_j \to t_j' = t_i t_j \tag{2.7}$$

Pour connaître le circuit de Clifford permettant de transformer le générateur de cette manière, il suffit de noter que l'Eq. (2.7) agit comme $\Lambda_j(X_i)$, mais sur les $\{s, t\}$ plutôt que sur les $\{Z, X\}$, cf. Fig. 2.1(c-f). En d'autres mots, la porte encodée $\mathcal{C}\Lambda_j(X_i)\mathcal{C}^\dagger$ effectue cette transformation du générateur. Nous nous retrouvons donc avec l'Eq. (2.8).

$$s_i' = \left[\mathcal{C}\Lambda_j(X_i)\mathcal{C}^\dagger\right]\left[\mathcal{C}Z_i\mathcal{C}^\dagger\right]\left[\mathcal{C}\Lambda_j(X_i)\mathcal{C}^\dagger\right]^\dagger \qquad t_j' = \left[\mathcal{C}\Lambda_j(X_i)\mathcal{C}^\dagger\right]\left[\mathcal{C}X_j\mathcal{C}^\dagger\right]\left[\mathcal{C}\Lambda_j(X_i)\mathcal{C}^\dagger\right]^\dagger \tag{2.8}$$

$$s_i' = \left[\mathcal{C}\Lambda_j(X_i)\right]Z_i\left[\mathcal{C}\Lambda_j(X_i)\right]^\dagger \qquad\qquad t_i' = \left[\mathcal{C}\Lambda_j(X_i)\right]X_i\left[\mathcal{C}\Lambda_j(X_i)\right]^\dagger \tag{2.9}$$

L'Eq. (2.9) nous montre que le circuit $\mathcal{C}' = \mathcal{C}\Lambda_j(X_i)$ est le circuit d'encodage du code stabilisateur généré par les générateurs modifiés. Un raisonnement similaire peut être fait pour une deuxième (et dernière) opération élémentaire qui consiste à échanger un s_i et un t_i, c.-à-d. $s_i \leftrightarrow t_i$. Il suffit de noter que l'effet de cette transformation est le même que celui d'une porte H sur Z et X. Le circuit devient donc $\mathcal{C}' = \mathcal{C}H_i$ dans ce cas. Finalement, pour bâtir un circuit d'encodage à partir d'un générateur du stabilisateur seulement, il suffit d'appliquer itérativement des opérations élémentaires en partant du code trivial généré par $\{Z_i, X_j\}$. Pour chaque générateur, nous procédons de la manière suivante. En boucle, nous exprimons le générateur désiré en fonction de ceux obtenus jusqu'alors et nous choisissons arbitrairement une opération élémentaire nous rapprochant de notre objectif. Une fois le générateur désiré obtenu, nous passons au générateur suivant et ainsi de suite jusqu'à ce que nous ayons construit le circuit complet. Cette procédure est loin d'être optimale, c.-à-d. qu'il pourrait exister des circuits moins longs, mais équivalents. Pour des codes faisant intervenir de petits nombres de qubits, comme c'est le cas dans l'article, cette façon de faire est suffisante. Nous renvoyons à [23] pour une méthode beaucoup plus efficace, mais peut-être moins intuitive.

2.5 Erratum

Le dénominateur de la deuxième et de la troisième équation de la section III devrait être sous une racine carrée tel que l'indique l'Eq. (2.10).

$$\frac{\cos^2\theta_0\,|0\rangle + \sin^2\theta_0\,|1\rangle}{\sqrt{\cos^4\theta_0 + \sin^4\theta_0}} \tag{2.10}$$

2.6 Article

Distillation of Non-Stabilizer States for Universal Quantum Computation

Guillaume Duclos-Cianci[1,*] and Krysta M. Svore[2,†]

[1] *Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1 (Canada)*
[2] *Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA 98052 (USA)*

(Dated: April 19, 2015)

Magic state distillation is a fundamental technique for realizing fault-tolerant universal quantum computing, and produces high-fidelity Clifford eigenstates, called magic states, which can be used to implement the non-Clifford $\pi/8$ gate. We propose an efficient protocol for distilling other *non-stabilizer* states that requires only Clifford operations, measurement, and magic states. One critical application of our protocol is efficiently and fault-tolerantly implementing arbitrary, non-Clifford, single-qubit rotations in, on average, *constant* online circuit depth and polylogarithmic (in precision) offline resource cost, resulting in significant improvements over state-of-the-art decomposition techniques. Finally, we show that our protocol is robust to noise in the resource states.

PACS numbers: 03.67.Lx, 03.67.Pp, 03.65.Fd
Keywords: state distillation, Solovay-Kitaev decomposition

I. INTRODUCTION

Given recent progress in quantum algorithms, quantum error correction, and quantum hardware, a *scalable* quantum computer is becoming closer and closer to reality. For many proposed quantum computer architectures, e.g., topological systems based on the braiding of non-Abelian anyons [1–3] or the surface-code model based on code deformation [4], Clifford operations, stabilizer-state preparations, and measurements can be implemented efficiently. However, these operations alone are not sufficient for quantum universality and can be simulated classically [5]. Magic state distillation [6–9] produces Clifford eigenstates, which in turn can be used to realize a non-Clifford operation, e.g., the single-qubit $\pi/8$ gate, T.

In this paper, we present an efficient protocol for distilling other *non-stabilizer* states. Our protocol uses only $|H\rangle$-type magic resource states, Clifford operations, and measurements, and is robust to noise in the resource states. One notable application of our protocol is producing an arbitrary single-qubit, fault-tolerant unitary operation. Previously, a single-qubit unitary U was decomposed into a discrete set of gates, typically $\{H, T\}$, using Solovay-Kitaev decomposition [10, 11], which efficiently produces an approximate fault-tolerant implementation of U with circuit depth $\Theta(\log^c(1/\epsilon))$, where ϵ is the precision and c is around 3.97 [11, 12]. Remarkably, efficient decomposition algorithms have recently been proposed which lower c to 1 [13, 14]. Each T gate in the decomposed sequence requires a number of copies of a quantum magic state $|H\rangle$, dependent on the specific state distillation protocol and purity of the state [6–9]. We show that our protocol requires, on average, only *constant* online circuit depth and fewer resources than state-of-the-art decomposition techniques, enabling a much cheaper (in number of qubits and gates) physical implementation.

In Section I, we review the two-qubit circuit necessary to apply a single-qubit rotation using a resource state. In Section II, we show that the same circuit can be used to distill a familiy of resource states. In Section III, we show how to efficiently use this family to implement an arbitrary single-qubit rotation efficiently. In Section IV, we show how to obtain tuneable tradeoffs between online and offline costs. Finally, in Appendix A, we show numerical evidence that our protocol is robust to small imperfections.

II. DISTILLING MAGIC STATES AND IMPLEMENTING ROTATIONS

We first review how to perform an arbitrary rotation about the Z-axis using a resource state, and in particular the T rotation. A state $|\psi\rangle$ is *magic* if we can "distill" a purer $|\psi\rangle$ state from a Clifford circuit applied to n noisy copies of $|\psi\rangle$. We focus on the $+1$ eigenstate of the Hadamard operation H, $|H\rangle = \cos\frac{\pi}{8}|0\rangle + \sin\frac{\pi}{8}|1\rangle$. We assume throughout that Clifford operations are perfect such that arbitrarily pure resource states can be obtained by applying a distillation protocol recursively [6–9]. We concentrate on single-qubit states found in either the XZ- or XY-plane of the Bloch sphere; note that a state can be rotated from one plane to the other through application of the Clifford $HS^\dagger H$ operation.

Consider states of the form $|Z(\theta)\rangle = |0\rangle + e^{i\theta}|1\rangle$ and $|\psi\rangle = a|0\rangle + b|1\rangle$. The circuit to implement a rotation around the Z-axis using $|Z(\theta)\rangle$ as a resource state is presented in Fig. 1. Upon measurement of the first qubit in the computational basis, we obtain either

$$\xrightarrow{m=0} a|0\rangle + be^{i\theta}|1\rangle, \text{ or}$$
$$\xrightarrow{m=1} ae^{i\theta}|0\rangle + b|1\rangle = a|0\rangle + be^{-i\theta}|1\rangle,$$

each with probability 1/2. Thus, the rotation angle is

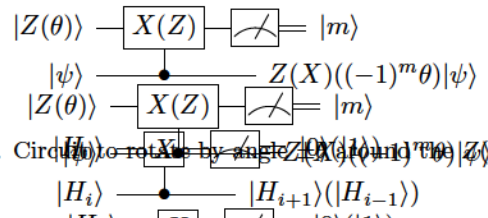


FIG. 1. Circuit to rotate by angle 2θ around the $Z(X)$-axis.

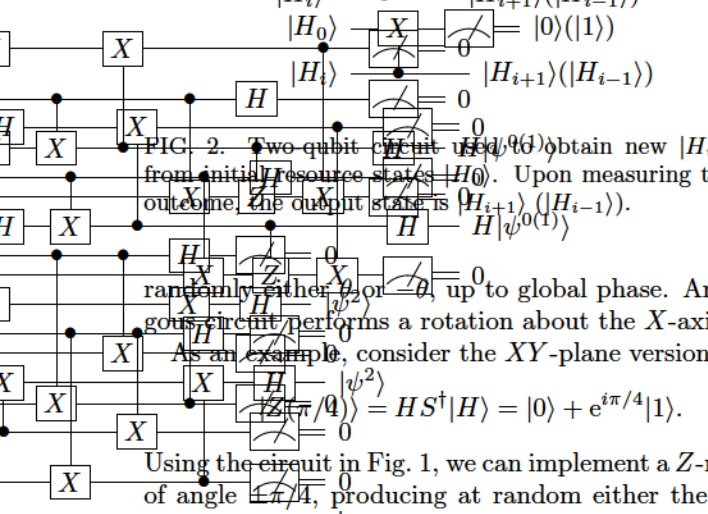


FIG. 2. Two qubit circuit used to obtain new $|H_i\rangle$ states from initial resource states $|H_0\rangle$. Upon measuring the 0 (1) outcome, the output state is $|H_{i+1}\rangle$ ($|H_{i-1}\rangle$).

randomly either θ or $-\theta$, up to global phase. An analogous circuit performs a rotation about the X-axis [4].

As an example, consider the XY-plane version of $|H\rangle$:

$$|\psi^2\rangle = HS^\dagger|H\rangle = |0\rangle + e^{i\pi/4}|1\rangle.$$

Using the circuit in Fig. 1, we can implement a Z-rotation of angle $\pm\pi/4$, producing at random either the T gate or its adjoint, $T^\dagger$. We can deterministically *correct* the angle by applying the phase gate S: $ST^\dagger|\psi\rangle = T|\psi\rangle$. For general rotations, deterministic correction is not possible using only Clifford gates.

III. DISTILLING OTHER NON-STABILIZER STATES

We now present our protocol for producing other non-stabilizer states using a very simple two-qubit Clifford circuit and $|H\rangle$ states as an initial resource.

Consider the circuit of Fig. 2. One can easily verify that it measures the parity of the two input qubits and decodes the resulting state into the second qubit. Consider the two inputs to be $|H\rangle$ states and define $\theta_0 = \frac{\pi}{8}$ and $|H\rangle = |H_0\rangle = \cos\theta_0|0\rangle + \sin\theta_0|1\rangle$. Then upon application of the controlled-NOT gate $\Lambda(X)$,

$$|H_0\rangle|H_0\rangle \xrightarrow{\Lambda(X)} \cos^2\theta_0|00\rangle + \sin^2\theta_0|01\rangle + \cos\theta_0\sin\theta_0(|11\rangle + |10\rangle).$$

Upon measurement m of the first qubit, we have

$$\xrightarrow{m=0} \frac{\cos^2\theta_0|0\rangle + \sin^2\theta_0|1\rangle}{\cos^4\theta_0 + \sin^4\theta_0}, \text{ or } \xrightarrow{m=1} \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

We define θ_1 such that

$$\cos\theta_1|0\rangle + \sin\theta_1|1\rangle = \frac{\cos\theta_0^2|0\rangle + \sin\theta_0^2|1\rangle}{\cos^4\theta_0 + \sin^4\theta_0},$$

from which we deduce $\cot\theta_1 = \cot^2\theta_0$. We define $|H_1\rangle = \cos\theta_1|0\rangle + \sin\theta_1|1\rangle$, a non-stabilizer state obtained from $|H\rangle$ states, Clifford operations, and measurements. If the measurement outcome is 1, then we obtain a stabilizer state and discard the output (see Fig. 2). The measurement outcomes occur with respective probabilities $p_0 = \cos^4\theta_0 + \sin^4\theta_0 = \frac{3}{4}$ and $p_1 = 1 - p_0 = \frac{1}{4}$.

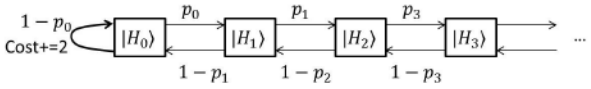


FIG. 3. Obtaining non-stabilizer states from initial $|H\rangle$ states. Using $|H_i\rangle$ and $|H_0\rangle$ states probabilistically yields a $|H_{i-1}\rangle$ or $|H_{i+1}\rangle$ using the circuit of Fig. 2. Each ladder step costs one $|H_0\rangle$ state, except the first one which costs two.

We now recurse on this protocol using the non-stabilizer states produced by the previous round of the protocol as input to the circuit in Fig. 2. We define $|H_i\rangle = \cos\theta_i|0\rangle + \sin\theta_i|1\rangle$, where $\cot\theta_i = \cot^{i+1}\theta_0$. Using as input the previously produced $|H_i\rangle$ state and a new $|H_0\rangle$ state, we have

$$|H_0\rangle|H_i\rangle \xrightarrow{\Lambda(X)} \cos\theta_0\cos\theta_i|00\rangle + \sin\theta_0\sin\theta_i|01\rangle + \sin\theta_0\cos\theta_i|10\rangle + \cos\theta_0\sin\theta_i|11\rangle.$$

Upon measurement of the first qubit, we have

$$\begin{aligned} &\xrightarrow{m=0} (\cos\theta'|0\rangle + \sin\theta'|1\rangle), \\ &\xrightarrow{m=1} (\cos\theta''|0\rangle + \sin\theta''|1\rangle), \text{ where} \\ \cot\theta' &= \cot\theta_i\cot\theta_0 = \cot^{i+2}\theta_0 = \cot\theta_{i+1}, \\ \cot\theta'' &= \cot\theta_i\tan\theta_0 = \cot^{i}\theta_0 = \cot\theta_{i-1}. \end{aligned}$$

Thus, if we measure $m = 0$, we obtain the state $|H_{i+1}\rangle$ and if we measure $m = 1$, we obtain $|H_{i-1}\rangle$. The probability of measuring 0 is given by

$$p_{0,i} = \cos^2\theta_i\cos^2\theta_0 + \sin^2\theta_i\sin^2\theta_0.$$

Note that $\frac{3}{4} \le p_{0,i} < \cos^2\frac{\pi}{8} = 0.853\ldots$

We can view this recursive process as a semi-infinite random walk with biased non-homogeneous probabilities, as Fig. 3 illustrates. Every time a step is taken along this "ladder" of states, one $|H\rangle \equiv |H_0\rangle$ is consumed, except at the first step of the ladder when we require two $|H\rangle$ states; if $m = 1$ at the first node, we discard the output and start with two new $|H\rangle$ states.

We can produce a denser ladder of states by using additional resource states $|\psi_0^{0,1,2}\rangle$. The Clifford circuit given in Fig. 4(a) takes as input four $|H\rangle$ states. It measures the stabilizer code given in Table I(a). With probability $3(2+\sqrt{2})/32 \approx 0.320$, the measurement outcome is 000 and the resulting state is $|\psi_0^0\rangle = \cos\phi_0^0|0\rangle + \sin\phi_0^0|1\rangle$ with $\phi_0^0 = \frac{\pi}{2} - \cot^{-1}\left(\frac{2+3\sqrt{2}}{6+5\sqrt{2}}\right) \approx 0.446$. Otherwise the output is discarded. Since the probability of success is 0.320 and every trial consumes four copies of $|H_0\rangle$, the average cost to produce $|\psi_0^0\rangle$ is 12.50 $|H_0\rangle$ states.

Another interesting state is obtained using the same circuit with one input state replaced with a $|+\rangle$ state.

(a)

S	$\pm$	0	1	2	3
s_0	+	X	Z	X	.
s_1	+	.	X	Z	X
s_2	+	X	.	X	Z
$\overline{Z}$	+	Z	Z	Z	Z

(b)

S	$\pm$	0	1	2	3
s_0	+	X	X	X	X
s_1	+	Z	.	Z	.
s_2	+	Z	.	.	Z
$\overline{Z}$	+	Z	Z	Z	Z

TABLE I. The stabilizer code decoded by the circuit of (a) Fig. 4(a) and (b) Fig. 4(b).

Measurement 000 is obtained with probability $(6+\sqrt{2})/32 \approx 0.232$, resulting in the state $|\psi_0^1\rangle = \cos\phi_0^1|0\rangle + \sin\phi_0^1|1\rangle$ with $\phi_0^1 = \frac{\pi}{2} - \cot^{-1}\left(\frac{2\sqrt{2}}{3+\sqrt{2}}\right) \approx 0.570$. Since the probability of success is 0.232 and every trial consumes three $|H_0\rangle$ states, the average cost to produce $|\psi_0^1\rangle$ is 12.95 $|H_0\rangle$ states. Fig. 4(a) shows a circuit which produces the output state $|\psi_0^2\rangle = \cos\phi_0^2|0\rangle + \sin\phi_0^2|1\rangle$ with $\phi_0^2 = \frac{\pi}{2} - \cot^{-1}\left(\frac{7}{6\sqrt{2}}\right) \approx 0.690$, when measurement 000 is obtained (with probability $11/32 \approx 0.344$). The probability of success is 0.344 and the average cost to produce $|\psi_0^2\rangle$ is 11.64 $|H_0\rangle$ states.

Now we can use one of these non-stabilizer states as input to the circuit in Fig. 2 in place of the top $|H_0\rangle$ state. Begin with states $|\psi_0^i\rangle$ and $|H_0\rangle$. If $m=1$, the state is discarded. Otherwise, we obtain $|\psi_1^i\rangle = \cos\phi_1^i|0\rangle + \sin\phi_1^i|1\rangle$, where $\cot\phi_1^i = \cot\phi_0^i\cot\theta_0$. As before, we define $|\psi_i^j\rangle = \cos\phi_i^j|0\rangle + \sin\phi_i^j|1\rangle$, where $\cot\phi_i^j = \cot\phi_0^j\cot^i\theta_0$. If we input states $|\psi_i^j\rangle$ and $|H_0\rangle$, we obtain

$$|H_0\rangle|\psi_i^j\rangle \xrightarrow{\Lambda(X)} \cos\theta_0\cos\phi_i^j|00\rangle + \sin\theta_0\sin\phi_i^j|01\rangle + \sin\theta_0\cos\phi_i^j|10\rangle + \cos\theta_0\sin\phi_i^j|11\rangle,$$

such that the output state is, depending on the measurement outcome,

$$\xrightarrow{m=0} |\psi_{i+1}^j\rangle, \qquad \text{or} \quad \xrightarrow{m=1} |\psi_{i-1}^j\rangle.$$

Denser "ladders" of states can be obtained using $|\psi_0^{0,1,2}\rangle$ as inputs in place of the top $|H_0\rangle$ state.

A priori, noise in the $|H_0\rangle$ resource states could be amplified by the circuit in Fig. 2 and affect the purity of the $|H_i\rangle$ states. However, this is not the case, and in fact an improvement in online and offline costs can be obtained since our circuits allow noisier $|H\rangle$ states to be used at the first step of the ladder; see appendix A for details.

IV. APPLICATION TO SINGLE-QUBIT ROTATIONS

We now show how to use the ladders of states to enable the fault-tolerant approximation of any single-qubit rotation. Results do not include these improvements in offline cost, so an additional gain factor between 2 and 10, depending on ϵ, is expected. Recall the circuit given in Fig. 1. If we input either $HS^\dagger H|H_i\rangle$ or $HS^\dagger H|\psi_i^j\rangle$ in place of the top qubit, we obtain rotation $Z(\pm 2\theta_i)$ on $|\psi\rangle$. Note that there is a factor of two difference between the angle θ_i involved in the description of the state and the rotation applied, e.g., the $|H_0\rangle$ state is over $\theta_0 = \frac{\pi}{8}$, and can be used to implement a $\frac{\pi}{4}$ rotation. Also, since $0 < \theta_i < \frac{\pi}{4}$ $(\forall i)$, the discontinuity of cotangent is not a problem.

Although the circuit in Fig. 1 randomly applies $\pm\theta$, our protocols still result in efficient application of the desired Z-rotation. We propose the following protocol to approximate a Z-rotation $Z(\phi)$:

1. Set desired accuracy ϵ.
2. Pick a target rotation angle $0 < \phi < 2\pi$.
3. Find the state $|H_i\rangle$ (or denser state $|\psi_i^j\rangle$) such that $2\theta_i$ is close to ϕ.
4. Simulate an instance of the ladder to obtain that state and add its cost to the offline cost.
5. Apply a rotation using $|H_i\rangle$ (or denser state $|\psi_i^j\rangle$) as input to the circuit of Fig. 1 and add one to the online cost.
6. Recurse on steps 3 through 5 until the desired accuracy is reached.

Thus, one has to implement a sequence of j rotations $\{Z(2\theta_{i_j})\}$ on $|\psi\rangle$ using the sequence of states $\{|H_{i_j}\rangle\}$, such that $Z(\phi) \approx \prod_j Z(2\theta_{i_j})$. The number of rotations required is given by $\left|\{|H_{i_j}\rangle\}\right|$ (i.e., online cost, see below).

We define the accuracy of the applied rotation V compared to the target rotation $U = Z(\phi)$ as

$$\max_{|\psi\rangle} D(U|\psi\rangle\langle\psi|U^\dagger, V|\psi\rangle\langle\psi|V^\dagger),$$

where $D(\rho,\sigma)$ is the trace distance between states ρ and σ. If U and V are rotations about the same axis, which is our case, one can show that for small angles of rotation, this reduces to the difference of rotation angles: $\epsilon = \Delta\phi$. In [12], the distance measure used is $D(U,V) = \sqrt{(2-|\mathrm{tr}(UV^\dagger)|)/2}$. In the case of rotations about the same axis, it can be reduced to $\sqrt{1-|\cos(\Delta\phi)|} \approx \Delta\phi/\sqrt{2}$ for small $\Delta\phi$.

We define an online and offline cost to apply a unitary gate. The *online cost*, C_{on}, is the expected number of $|H_i\rangle$ states required to implement the unitary. The *offline cost*, C_{off}, is the total number of distilled $|H_0\rangle$ states required to obtain all of the intermediate $|H_i\rangle$ states used to perform the given unitary. For Solovay-Kitaev decomposition, the offline cost equals the online cost and is given by the total number of T and $T^\dagger$ gates in the decomposition. In both cases, we do not count the cost of initially distilling $|H_0\rangle$ states.

We simulated $\sim 1.8\times10^4$ instances of our $|H\rangle$ protocol, each for a random angle ϕ and target accuracy between

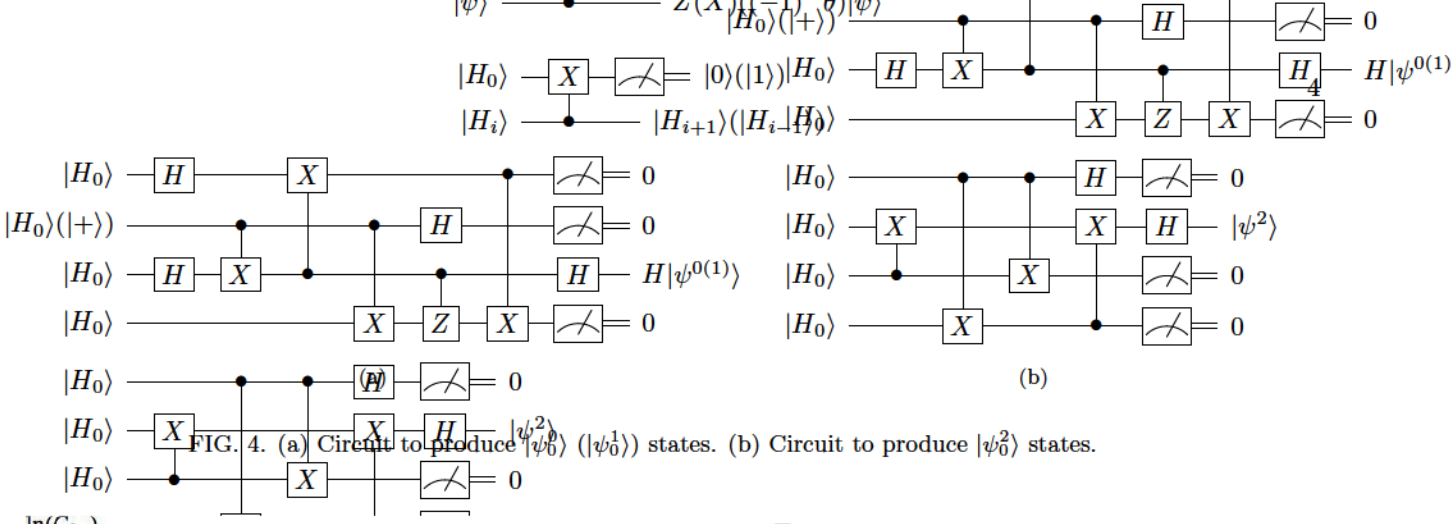


FIG. 4. (a) Circuit to produce $|\psi_0^0\rangle$ ($|\psi_0^1\rangle$) states. (b) Circuit to produce $|\psi_0^2\rangle$ states.

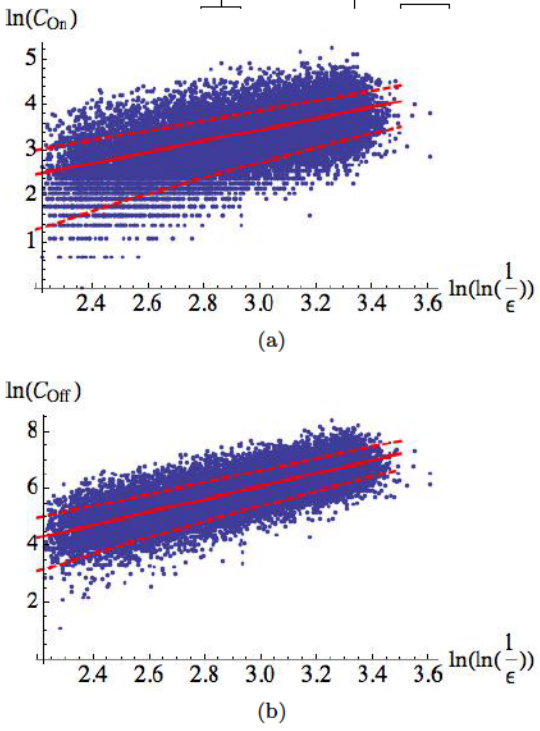


FIG. 5. (Color online) Simulation of target accuracies for random angles. (a) Full line: Fit of the average, $\ln(C_{\text{on}}) = -0.21+1.23\ln(\ln(1/\epsilon))$. Dashed lines: standard deviation around the mean, $\ln(\Delta C_{\text{on}}) = -0.30+0.83\ln(\ln(1/\epsilon))$. (b) Full line: Fit of the average, $\ln(C_{\text{on}}) = -0.44 + 2.22\ln(\ln(1/\epsilon))$. Dashed lines: standard deviation around the mean, $\ln(\Delta C_{\text{on}}) = 0.02 + 1.87\ln(\ln(1/\epsilon))$.

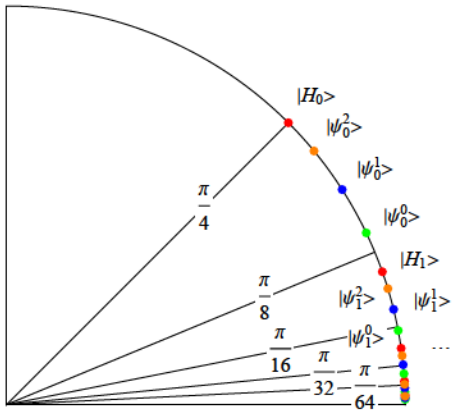


FIG. 6. (Color online) Dots: Rotation angles achievable using the $|H_i\rangle$, $|\psi_i^0\rangle$, $|\psi_i^1\rangle$, $|\psi_i^2\rangle$ protocol, respectively.

1

$10^{-12} < \epsilon < 10^{-4}$. We assume that $C_{\text{on}} \sim \ln_{\text{on}}^{c}(\frac{1}{\epsilon})$, and $C_{\text{off}} \sim \ln^{c_{\text{off}}}(\frac{1}{\epsilon})$, where C_{on} and C_{off} are the online and offline costs, respectively, such that $\ln C_{\text{on}} \sim c_{\text{on}} \ln\ln(\frac{1}{\epsilon})$, and $\ln C_{\text{off}} \sim c_{\text{off}} \ln\ln(\frac{1}{\epsilon})$. The results are given in Fig. 5. From linear fits to the data, we find $\ln(C_{\text{on}}) = -0.21+1.23\ln(\ln(1/\epsilon))$ with a standard deviation around the mean of $\ln(\Delta C_{\text{on}}) = -0.30 + 0.83\ln(\ln(1/\epsilon))$, and $\ln(C_{\text{on}}) = -0.44 + 2.22\ln(\ln(1/\epsilon))$ with a standard deviation around the mean of $\ln(\Delta C_{\text{on}}) = 0.02 + 1.87\ln(\ln(1/\epsilon))$. We deduce that $c_{\text{on}} \sim 1.23$ and $c_{\text{off}} \sim 2.22$ for our protocol.

For the denser protocol, the offline costs are 12.50, 12.95, and 11.64 for $|\psi_0^0\rangle$, $|\psi_0^1\rangle$, and $|\psi_0^2\rangle$, respectively. The denser set of states results in improved scalings for both the online and offline costs: $c'_{\text{on}} \sim 1.04$ and $c'_{\text{off}} \sim 1.64$, where $'$ denotes the denser protocol. However, the offline costs of our new states $|\psi_0^i\rangle$ are improved only when precisions are smaller then $\epsilon \approx 1.28 \times 10^{-5}$. Fig. 6 shows the angles that are obtainable using our four protocols.

Fig. 7 shows the behavior of the protocols on Z rotations and arbitrary rotations. For an arbitrary rotation, recall that a single-qubit unitary U is composed of three rotations around the X- and Z-axes [15]: $U \propto X(\alpha)Z(\beta)X(\gamma)$, for some angles α, β, γ. We can use our protocol to implement both Z and X rotations as previously outlined. Fig. 7(a) plots the fit for Solovay-Kitaev decomposition [12] (solid line), the online cost (dashed), and offline cost (dotted). For all practical precisions, the online cost of our proposed scheme is consistently smallest. The offline cost is advantageous when $\epsilon \leq 4.41\times10^{-4}$ for Z-rotations and $\epsilon < 1.03\times10^{-6}$ for random unitaries.

V. MINIMIZING ONLINE COST

We can further minimize the online cost by considering instead the following protocol to implement a Z ro-

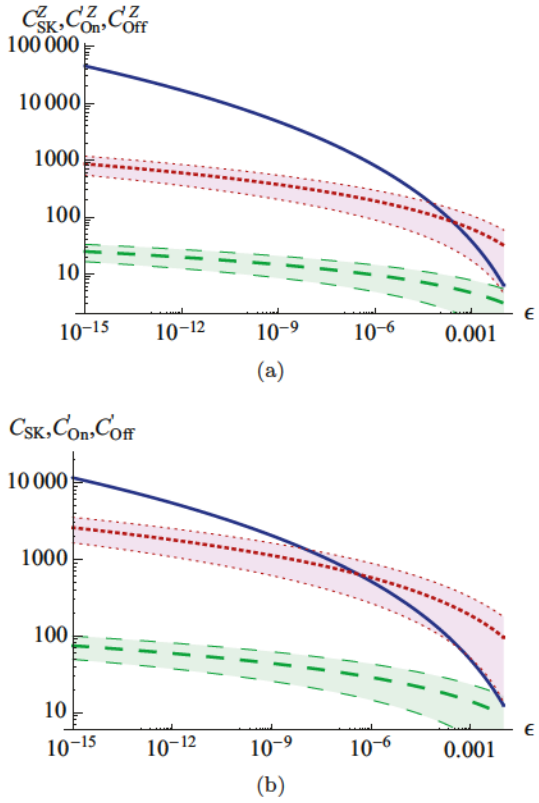


FIG. 7. (Color online) Cost of (a) random Z-rotations and (b) random unitaries as a function of precision ϵ. Solid line: SK decomposition [12]. Dotted line: Offline cost using $|H\rangle$, or $\{|\psi^0\rangle, |\psi^1\rangle, |\psi^2\rangle\}$ as initial resources. Dashed line: Online cost using $|H\rangle$, or $|\psi^0\rangle, |\psi^1\rangle, |\psi^2\rangle\}$ as initial resources. The shaded regions around the dashed and dotted lines represent the standard deviation around the mean. (a) $\ln(C'^Z_{\rm SK}) = -4.88 + 4.41\ln(\ln(1/\epsilon))$; $\ln(C'^Z_{\rm On}) = -0.46 + 1.04\ln(\ln(1/\epsilon))$; $\ln(C'^Z_{\rm Off}) = 0.96 + 1.64\ln(\ln(1/\epsilon))$. (b) $\ln(C'_{\rm SK}) = -2.67 + 3.40\ln(\ln(1/\epsilon))$; $\ln(C'_{\rm On}) = -0.46 + 1.04\ln(\ln(1/\epsilon)) + \ln 3$; $\ln(C'_{\rm Off}) = 0.96 + 1.64\ln(\ln(1/\epsilon)) + \ln 3$.

tation by angle ϕ: Prepare *offline* the state $|Z(\phi)\rangle$ using the protocol described to apply $|Z(\phi)\rangle$ to a $|0\rangle$ ancilla. Then, use $|Z(\phi)\rangle$ *online* to apply the rotation to the desired qubit. With probability $\frac{1}{2}$, the rotation $Z(\phi)$ is applied and the online cost is 1. If it fails, prepare *offline* $|Z(2\phi)\rangle$; with probability $\frac{1}{2}$, $Z(\phi)$ is applied online and the online cost is 2. If it fails, prepare *offline* $|Z(4\phi)\rangle$, and so on. The probability that the procedure requires exactly n iterations decreases exponentially with n; the process is a negative binomial of parameter $p = \frac{1}{2}$ and the expected number of online rotations for success is $\sim \frac{1}{p} = 2$. We simulated this process for random angles $0 < \phi < 2\pi$ and accuracies $10^{-12} < \epsilon < 10^{-4}$ and found the expected number of online rotations is $\langle C''_{\rm on}\rangle = 1.99$ and the offline cost is $c''_{\rm off} \sim 1.75$. Note that any method can be used to prepare the ancilla state offline, and here we use our protocol for preparation. We discovered after writing that a similar technique was described in [16].

θ	C	$\epsilon = 10^{-4}$	$\epsilon = 10^{-8}$	$\epsilon = 10^{-12}$
$\pi/16$	C_{SK}	43.83	2646	29120
	$C_{\rm on}$	10.20	24.52	41.95
	$C'_{\rm on}$	5.88	12.48	19.38
	$C_{\rm off}$	73.06	349.8	874.4
	$C'_{\rm off}$	98.29	306.1	595.0
$\pi/128$	C_{SK}	53.84	2879	29530
	$C_{\rm on}$	5.47	18.96	39.27
	$C'_{\rm on}$	3.32	9.27	16.91
	$C_{\rm off}$	49.18	313.0	923.9
	$C'_{\rm off}$	52.60	234.1	560.8
$\pi/1024$	$C_{\rm SK}$	128.1	2594	15075
	$C_{\rm on}$	7.99	23.08	42.93
	$C'_{\rm on}$	3.00	8.37	15.23
	$C_{\rm off}$	77.42	381.3	969.1
	$C'_{\rm off}$	65.75	245.5	530.7

TABLE II. $C_{\rm on}$ and $C_{\rm off}$ are online and offline costs using only $|H\rangle$ states, to precision ϵ. $C'_{\rm on}$ and $C'_{\rm off}$ refer to the costs for the denser protocol. $C_{\rm SK}$ is the extrapolated cost using [12].

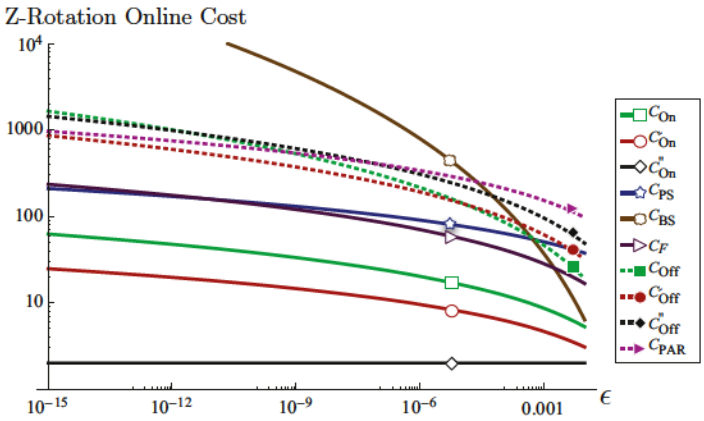


FIG. 8. (Color online) Comparison of online (solid) and offline (dashed) costs to decompose Z rotations vs. accuracy ϵ. Methods plotted include $C_{\rm PAR}$ [16], $C_{\rm PS}$ [13], $C_{\rm BS}$ [12], $C_{\rm F}$ [17]. $C_{\rm On}$, $C_{\rm Off}$, $C'_{\rm On}$, $C'_{\rm Off}$, $C''_{\rm On}$, $C''_{\rm Off}$ represent our $|H\rangle$ ladder, dense ladder, and minimal online cost with the dense ladder, respectively. The offline costs for $C_{\{\rm BS,PS,F\}}$ are equal to their online costs.

Table II lists the expected cost of Solovay-Kitaev decomposition [12] compared to our protocols for Z-rotations of angles $C_{\rm on}$ and $C_{\rm off}$ are the online and offline costs using only $|H\rangle$ states, $C'_{\rm on}$ and $C'_{\rm off}$ refer to the costs for the denser protocol, and $C_{\rm SK}$ is the extrapolated cost averaged over all unitaries [12]. In all cases, the online cost is minimal when our proposed scheme enhanced by $\{|\psi_0^0\rangle, |\psi_0^1\rangle, |\psi_0^2\rangle\}$ is used. For rougher precision, e.g., 10^{-4}, a Solovay-Kitaev decomposition may be desirable, while for finer precision, e.g., 10^{-8} or 10^{-12}, the cost of the Solovay-Kitaev decomposition becomes prohibitive, indicating the necessity and significant advantage of our proposed protocol.

Fig. 8 compares the cost of state-of-the-art decomposition techniques with our protocols. The plot highlights the tradeoffs between the various methods. Note that we only plotted two methods, our protocol C'' and C_{PAR} (which uses C_{F} to prepare the state), using the minimal online framework, but the other techniques could also be used to prepare the state offline, yielding an expected online cost of 2 and a roughly doubled offline cost. Our protocols C, C', and C'' exhibit a very clear tradeoff between online circuit depth and offline cost. For example, if operations on logical qubits must be minimized (due to noise), then trading offline resources for low online circuit depth is desirable, making C, C', and C'' advantageous compared to $C_{\{\mathrm{BS,F,PS}\}}$. C'' is competitive with the minimal-online versions of C_{F} (plotted as C_{PAR}) and C_{PS} (not plotted). In practice, several decomposition techniques will be used throughout the compilation of a quantum algorithm.

Finally, our protocol can be used to fault-tolerantly implement elements of the V basis, which consists of $V_{\{1,2,3\}} = (I+2i\{X,Y,Z\})/\sqrt{5}$ and their inverses. The V basis was shown to be *efficiently universal*, guaranteeing decompositions to be of depth $O(\log(1/\epsilon))$ [18]. It was previously dismissed as a candidate basis for decomposition due to the inability to implement the gates fault-tolerantly. However, our protocol enables fault-tolerant implementation: $V = Z(\pi/4)Z(2\theta_2)$, which is a T gate followed by a rotation using the $|H_2\rangle$ resource state (see Appendix of [19]). This has prompted the development of decomposition algorithms targeted to the V basis that may outperform those for the $\{H,T\}$ basis [19].

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a protocol to distill non-stabilizer states efficiently using magic states, Clifford operations, and measurements. One application of our protocol is implementing arbitrary single-qubit rotations with lower resource cost than state-of-the-art decomposition methods and *constant* online circuit depth. However, our protocols and other decomposition techniques are not exclusive. Some unitaries may be better implemented using other decomposition methods, while our scheme may be better suited for Z-rotations, which are common among quantum algorithms.

An extension of our work is to study other stabilizer circuits as "ladders" of states, or to use SH eigenstates distilled using the protocols of [6, 8]. Another interesting extension is to formalize what types of states can be distilled using probabilistic circuit constructions. Finally, optimizing the sequence of angles required to implement the desired rotation, or determining when to use a given decomposition technique, will be a necessary component of any quantum compiler.

ACKNOWLEDGMENTS

We thank Alex Bocharov and Cody Jones for many useful discussions.

* Guillaume.Duclos-Cianci@USherbrooke.ca
† ksvore@microsoft.com

[1] A. Kitaev, Ann. Phys. **303** (2003), arxiv:quant-ph/9707021.
[2] M. H. Freedman, Proc. Natl. Acad. Sci. **95** (1998).
[3] M. H. Freedman, M. Larsen, and Z. Wang, Commun. Math. Phys. **227** (2002), arxiv:quant-ph/0001108.
[4] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).
[5] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004), arXiv:quant-ph/0406196.
[6] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[7] A. M. Meier, B. Eastin, and E. Knill, "Magic state distillation with the four-qubit code," (2012).
[8] S. Bravyi and J. Haah, "Magic state distillation with low overhead," (2012), 1209.2426.
[9] N. C. Jones, "Multilevel distillation of magic states for quantum computing," (2012), 1210.3388.
[10] A. Kitaev et al., *Classical and Quantum Computation* (American Mathematical Society, Providence, RI, 2002).
[11] C. M. Dawson and M. A. Nielsen, "The Solovay-Kitaev algorithm," (2005), arxiv:quant-ph/0505030.
[12] A. Bocharov and K. M. Svore, Phys. Rev. Lett. **109**, 190501 (2012).
[13] P. Selinger, "Efficient clifford+T approximation of single-qubit operators," (2012), 1212.6253.
[14] V. Kliuchnikov, D. Maslov, and M. Mosca, "Practical approximation of single-qubit unitaries by single-qubit quantum clifford and T circuits," (2012), 1212.6964.
[15] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2000).
[16] N. C. Jones, J. D. Whitfield, P. L. McMahon, M.-H. Yung, R. V. Meter, A. Aspuru-Guzik, and Y. Yamamoto, "Simulating chemistry efficiently on fault-tolerant quantum computers," (2012), 1204.0567.
[17] A. Fowler, Quantum Information and Computation **11**, 867 (2011), quant-ph/0411206.
[18] A. W. Harrow, B. Recht, and I. L. Chuang, J. Math. Phys. **43** (2002).
[19] A. Bocharov, Y. Gurevich, and K. M. Svore, to appear in Phys. Rev. A (2013) quant-ph/1303.1411.

Appendix A: Noisy States

A priori, noise in the $|H_0\rangle$ resource states could be amplified by the circuit in Fig. 2 and affect the purity of the $|H_i\rangle$ states. However, we show this is not the case. We measure the accuracy of the imperfect $|H_i\rangle$ states using the trace distance on states ρ and σ: $D(\rho,\sigma) = \mathrm{tr}(\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)})/2$. We assume errors only occur on the $|H_0\rangle$ states. We numerically study three types

of errors. For the first error, we assume that the mixed state, ρ_0^a, is on the line joining the center of the Bloch sphere and the the perfect state, i.e.,

$$\rho_0^a(p) = (1-p)|H_0\rangle\langle H_0| + p|-H_0\rangle\langle -H_0|,$$

where $|-H_0\rangle = \sin\frac{\pi}{8}|0\rangle - \cos\frac{\pi}{8}|1\rangle$ is the state orthogonal to $|H_0\rangle$. We denote the imperfect version of $|H_i\rangle$ obtained from ρ_0^a states as ρ_i^a. We can always bring any mixed state into this form using twirling [7]. For the protocol to be practical, we require it to remain stable under the two following types of errors, where we assume the state is pure and the rotation is off of the desired axis by δ:

$$\rho_0^b(\delta) = \frac{1}{2}\left(I + \sin\left(\frac{\pi}{4}+\delta\right)X + \cos\left(\frac{\pi}{4}+\delta\right)Z\right),$$
$$\rho_0^c(\delta) = \frac{1}{2}\left(I + \sin\frac{\pi}{4}\cos\delta X + \sin\frac{\pi}{4}\sin\delta Y + \cos\frac{\pi}{4}Z\right).$$

We numerically generated 1000 pseudo-random instances of the protocol to produce $|H_i\rangle$ states for different values of i for each error type and for noise strengths $10^{-4}, 10^{-6}$, and 10^{-8}. Figure 9(a) shows an exponential decay of the distance between erroneous and ideal states; if we start with a $|H_0\rangle$ state distilled to our target accuracy, all subsequent derived $|H_i\rangle$ states will also be distilled to that accuracy. This further suggests that for larger values of i, noisier $|H_0\rangle$ states could be used to still achieve the desired accuracy, and in turn decrease the number of distillation recursions (and resources) necessary to prepare the $|H_0\rangle$ states.

Extrapolating from Fig.9(a), one could for example prepare ρ_{12} states with accuracy 10^{-9} using only input $|H_0\rangle$ states of accuracy 10^{-6}, saving at least one round of distillation prior to our protocol, reducing the total offline cost (including magic state distillation). Using states as noisy as possible and using the costs and accuracies presented in Table I of [7], we were able to estimate, via numerical simulations, the improvement factor to be gained in offline cost for different rotations and precisions. The results are presented in Fig. 9(b). Two important behaviors are noted. First, for any given realtive precision ϵ/ϕ, the improvement factor increases as the absolute precision ϵ goes down. Second, and more importantly, there is as much as an order of magnitude to be gained for rotation angles that are comparable to the desired accuracy ϵ, e.g., for $\epsilon = 5\times10^{-10}$ and $\phi \sim 100\epsilon$, there is a factor ~ 11 reduction in resource offline cost.

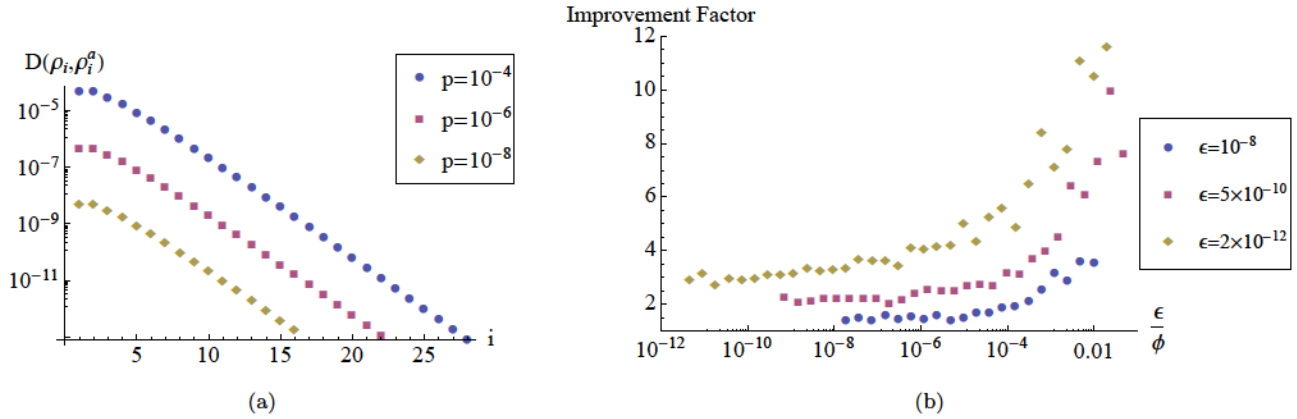


FIG. 9. (a) Evolution of the trace distance between imperfect ρ_i^a and perfect $|H_i\rangle$ states with noise p. Exponential decay fits give $(2.08 * 10^{-3}) \times 2.31^{-i}$, $(1.63 * 10^{-5}) \times 2.28^{-i}$ and $(1.26 * 10^{-7}) \times 2.24^{-i}$ for the circle, square and diamond data set, respectively. (b) Improvement factor of the total offline cost using the noisiest $|H_0\rangle$ states to distill $|H_i\rangle$ states of precision ϵ as a function of the relative precision of the rotation ϵ/ϕ.

2.7 Complément de résultats

Notez bien qu'il est préférable d'avoir lu l'article avant de lire cette section.

2.7.1 Coûts des états de l'échelle

Dans cette sous-section, nous fournissons une estimation du coût moyen de production des états de l'échelle à partir d'une banque d'états $|H\rangle$ et à l'aide du circuit de préparation présenté à la figure 2 de l'article. La Fig. 2.2 présente ce coût moyen, plus ou moins un écart-type. Le coût s'exprime en nombre d'états $|H\rangle$ requis. Le Tab. 2.1 présente les régressions linéaires des résultats. Notre analyse ne nous permet pas de conclure qu'une droite de la forme $mx + b$ soit véritablement la courbe vers laquelle ces données convergent. Toutefois, l'accord avec les résultats est suffisant pour le genre d'estimé que nous désirons accomplir ici. Un calcul détaillé impliquerait une marche aléatoire avec des frontières absorbantes et des probabilités de transferts dépendantes de i.

2.7.2 Coûts associés à l'échelle « densifiée »

Dans cette sous-section, nous fournissons les résultats de l'estimation du coût moyen pour appliquer des rotations d'angles aléatoires avec l'échelle densifiée à l'aide des états racines $\left|\psi^{0,1,2}\right\rangle$ présentés à la section III, plus précisément à la figure 4 et au tableau I de l'article. La Fig. 2.3 est l'analogue de la figure 5 de l'article pour C'_{on} et C'_{off}. Nous avons échantillonné environ 1.8×10^4 angles de rotations de manière uniformément aléatoire sur une échelle logarithmique pour des précisions cibles de $10^{-12} < \epsilon < 10^{-4}$. L'angle de rotation était toutefois toujours choisi de manière à être plus grand que la précision cible.

2.7.3 Analyse des erreurs

Dans cette sous-section, nous fournissons des résultats supplémentaires relatifs à l'appendice qui discute de l'effet des erreurs sur les états d'entrée $\rho_H \approx |H\rangle\langle H|$. Il n'est pas évident, à priori, que le protocole supprime les erreurs pouvant se retrouver sur les états ρ_H initiaux. Or, il semble que ce soit le cas. Dans le texte, les résultats pour seulement un type d'erreurs sont fournis. À la Fig. 2.4, nous présentons les données complétant celles fournies à la figure 9a de l'article. Nous rappelons aux Eqs. 2.11-2.13 les trois différents types

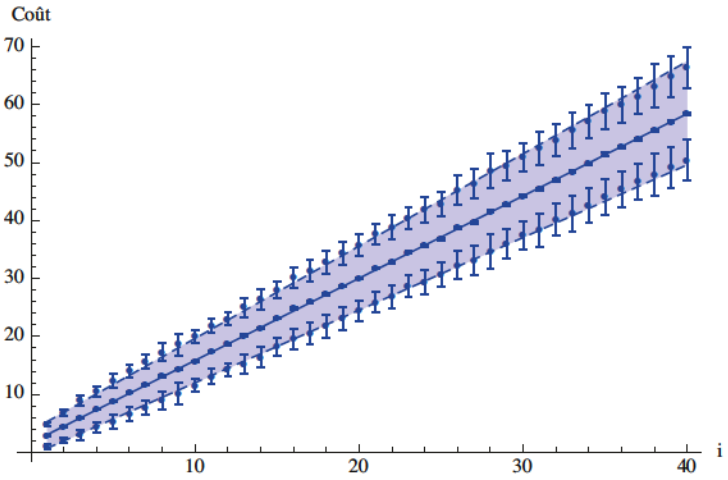
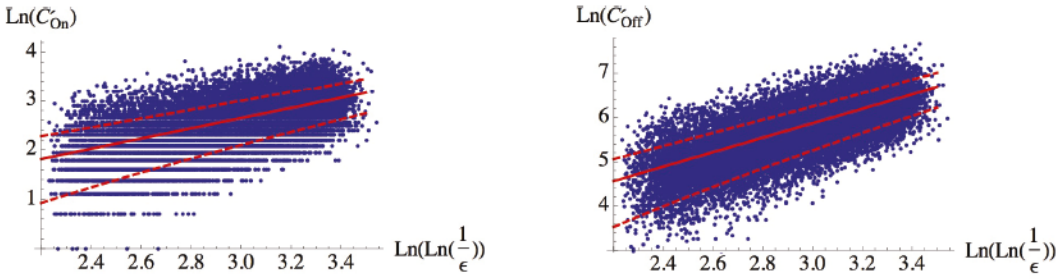


FIGURE 2.2 Coût, en nombre de $|H\rangle$, nécessaire à la production des états $|H_i\rangle$. Les points centraux donnent le coût moyen alors que ceux au-dessus et au-dessous donnent l'écart-type. La région ombragée représente donc l'intervalle de coût typique. Tous les points ont des barres d'erreurs représentant l'erreur statistique de l'échantillon de taille 1000 pour les valeurs $1 \leq i \leq 40$. L'erreur sur la moyenne est de l'ordre de la taille du point lui-même. L'erreur sur l'écart-type est naturellement plus importante.

Coût moyen	Écart-type
$1.67 + 1.42i$	$2.10 + 0.172i$

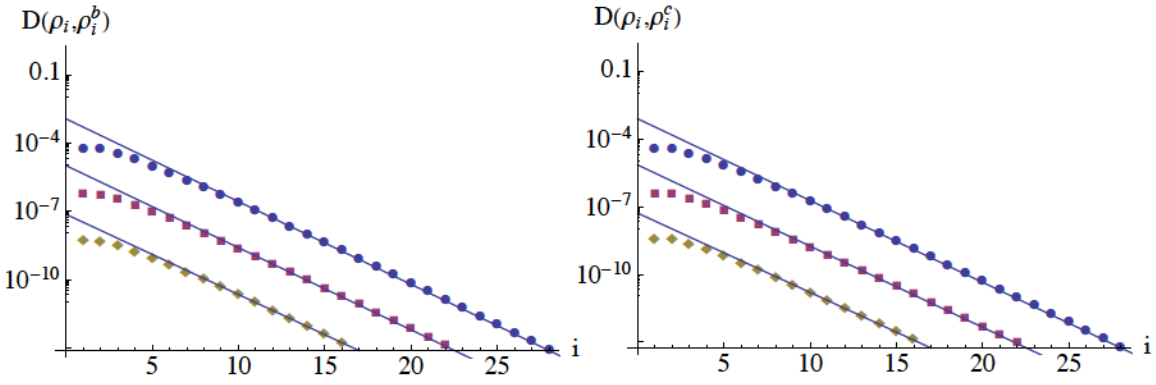
Tableau 2.1 Régressions linéaires des données de la Fig. 2.2.



(a) $\ln(C'_{\text{on}}) = -0.46 + 1.04 \ln\ln(1/\epsilon)$
$\ln(\Delta C'_{\text{on}}) = -0.072 + 0.619 \ln\ln(1/\epsilon)$

(b) $\ln(C'_{\text{off}}) = 0.96 + 1.64 \ln\ln(1/\epsilon)$
$\ln(\Delta C'_{\text{off}}) = 1.46 + 1.21 \ln\ln(1/\epsilon)$

FIGURE 2.3 Coûts direct (C'_{on}) et différé (C'_{off}) en fonction de la précision désirée ϵ.



(a) Erreurs du type Eq. (2.12). **(b)** Erreurs du type Eq. (2.13).

FIGURE 2.4 Suppression des erreurs « non-diagonales » par le circuit de production des états $|H_i\rangle$. L'axe horizontal, i, étiquette les états produits $|H_i\rangle$ et l'axe vertical donne la distance de trace entre ceux-ci et l'état idéal. Les trois courbes correspondent de haut en bas à $\delta = 10^{-4}$, 10^{-6} et 10^{-8}.

δ	erreurs a	erreurs b	erreurs c
10^{-4}	$(2.08 \times 10^{-3})2.31^{-i}$	$(1.17 \times 10^{-3})2.31^{-i}$	$(8.28 \times 10^{-4})2.31^{-i}$
10^{-6}	$(1.63 \times 10^{-5})2.28^{-i}$	$(1.03 \times 10^{-5})2.30^{-i}$	$(7.32 \times 10^{-6})2.30^{-i}$
10^{-8}	$(1.26 \times 10^{-7})2.24^{-i}$	$(7.50 \times 10^{-8})2.25^{-i}$	$(5.30 \times 10^{-8})2.25^{-i}$

Tableau 2.2 Décroissances exponentielles ajustées aux données pour les erreurs de types a, b et c, décrites aux Eqs. 2.12, 2.12 et 2.13, respectivement.

d'erreurs que nous avons étudiés.

$$\rho_a = \frac{1}{2}\left[\mathbb{1} + (1-2\delta)\left(\sin\frac{\pi}{4}X + \cos\frac{\pi}{4}Z\right)\right] \tag{2.11}$$

$$\rho_b = \frac{1}{2}\left[\mathbb{1} + \sin\left(\frac{\pi}{4}+\delta\right)X + \cos\left(\frac{\pi}{4}+\delta\right)Z\right] \tag{2.12}$$

$$\rho_c = \frac{1}{2}\left[\mathbb{1} + \sin\frac{\pi}{4}\cos(\delta)X + \sin\frac{\pi}{4}\sin(\delta)Y + \cos\frac{\pi}{4}Z\right] \tag{2.13}$$

Le Tab. 2.2 liste les différentes courbes exponentielles ajustées aux données. Comme les erreurs de tous les types sont supprimées par le circuit, nous concluons qu'il s'agit bien d'un circuit de distillation. Toutefois, la suppression est plutôt faible, car elle n'est qu'exponentielle. Rappelons que le coût de distillation l'est aussi. Il est donc nécessaire en général de prendre des états racines $|H\rangle$ de bonne qualité, c.-à-d. ayant été distillés à quelques reprises au préalable. De plus, notons que les erreurs de type a sont celles qui sont supprimées le plus lentement, c'est pourquoi nous n'avons inclus que celles-ci dans l'article.

Chapitre 3

Article : Reducing the quantum computing overhead with complex gate distillation

Guillaume Duclos-Cianci, David Poulin, *Reducing the quantum computing overhead with complex gate distillation*, Phys. Rev. A 91, 042315 (2015).

3.1 Contexte

Dans la dernière année de mon doctorat, j'ai cherché à pousser les idées sur lesquelles j'avais travaillé chez Microsoft Research. Dans cet article, nous étudions une famille d'états magiques récemment introduite par Landahl et Cesare [24] que j'ai indépendamment considérée peu après mon stage à l'été 2012. L'intérêt de cette famille est sa structure élégante, par opposition à celle introduite dans l'article précédent. J'ai découvert qu'un ingrédient crucial à leur distillation est la capacité d'appliquer une rotation d'angle π autour de l'axe défini sur la sphère de Bloch par l'état à distiller. Or, j'ai ensuite réalisé qu'on pouvait effectuer cette rotation pour un état à l'aide de celui qui le précède dans la famille. En me basant encore une fois sur un protocole de distillation existant, j'ai montré qu'on pouvait itérativement distiller tous ces états. À l'aide de cette famille étendue, j'ai ensuite numériquement montré que des gains significatifs étaient accomplis dans l'application de certaines portes. J'ai présenté ces résultats au groupe de John Preskill lors de ma visite à Caltech en décembre 2013 et à la rencontre de l'APS en 2014 (*March meeting*).

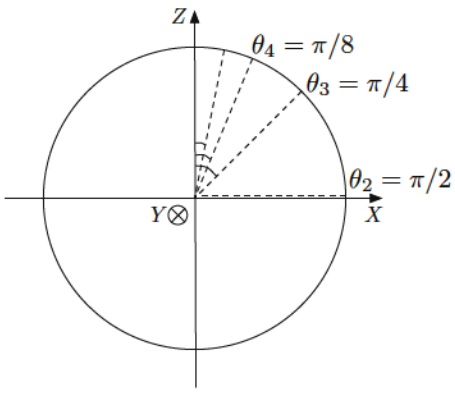


FIGURE 3.1 Quelques états de la famille $|Y_k\rangle$. Ils se retrouvent tous sur le méridien XZ de la sphère de Bloch.

3.2 Résumé

La section I de l'article discute du problème de goulot d'étranglement dans le cadre du calcul tolérant aux fautes. La section II introduit la famille d'états magiques et reformule le problème de la compilation dans ce contexte. Davantage de détails sont donnés dans la section 3.3 ci-dessous. La section III introduit le circuit de distillation et analyse ses performances. Certaines précisions sont apportées aux sections 3.4 et 3.5. La section IV discute d'un problème où l'utilisation de cette famille d'états est particulièrement appropriée : la simulation en chimie quantique. Elle propose aussi quelques façons d'améliorer encore davantage le protocole proposé. L'appendice A donne des détails supplémentaires concernant l'analyse d'erreurs et l'appendice B montre que la décomposition en angles d'Euler d'une petite rotation n'implique que de petits angles.

3.3 Famille d'états magiques

La famille d'états magiques que nous considérons s'exprime simplement : pour $k \geq 2$, posons $|Y_k\rangle = \cos(\theta_k/2)\,|0\rangle + \sin(\theta_k/2)\,|1\rangle$, où $\theta_k = 2\pi/2^k$. Tous ces états se retrouvent le long du méridien XZ de la sphère de Bloch comme le montre la Fig. 3.1. À l'aide de l'injection d'états, ceux-ci sont utilisés pour appliquer des rotations d'angle θ_k. Il se trouve que ces rotations permettent de définir à leur tour une famille d'opérateurs indispensables à leur propre distillation, opérateurs que nous notons W_k. Ils consistent en une rotation

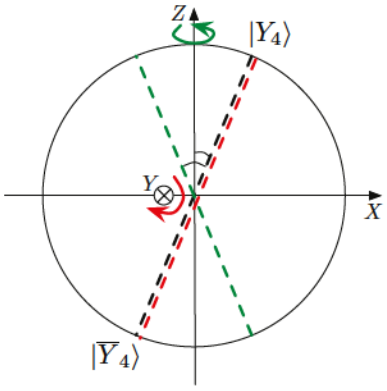


FIGURE 3.2 Exemple de rotation W_4. Nous appliquons d'abord la rotation Z d'angle π (en vert), qui correspond à une réflexion du plan XZ. Puis, nous appliquons la rotation Y d'angle $2\theta_4 = \theta_3$ à l'aide de l'état $|Y_3\rangle$ (en rouge).

d'angle π autour de l'axe défini par la base $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$. [1] Pour ce faire, nous appliquons d'abord la rotation Z d'angle π, qui correspond à une réflexion du plan XZ. Puis, nous appliquons la rotation Y d'angle $2\theta_k$ à l'aide de l'état $|Y_{k-1}\rangle$. La Fig. 3.2 illustre l'exemple $|Y_4\rangle$. Le calcul de l'Eq. (3.1) nous convainc que l'opérateur unitaire W_k est hermitien.

$$\begin{aligned} W_k^\dagger = (R_Y(2\theta_k)X)^\dagger &= XR_Y(-2\theta_k) \\ &= R_Y(2\theta_k)X \\ &= W_k \end{aligned} \tag{3.1}$$

Ses valeurs propres sont donc +1 et −1 et ces états propres $|Y_k\rangle$ et $|\overline{Y}_k\rangle$. Il s'agit en fait de l'opérateur phase dans la base correspondante.

3.4 Circuit de distillation

Les états magiques de la famille introduite ci-haut ne peuvent être préparés avec une précision arbitrairement grande. Or, la précision de ces états limite directement la précision des portes appliquées par leur injection. Dans le but de les rendre de plus en plus précis, nous les distillons. Cela est fait à l'aide du circuit présenté à la Fig. 3.3, circuit que nous avons conçu, inspirés par les travaux de Meier, Eastin et Knill [25]. Nous notons ρ_k une préparation imparfaite de $|Y_k\rangle$. Nous verrons à la section suivante pourquoi ce circuit permet en effet d'améliorer la précision des états d'entrée.

1. Dans ce chapitre la barre au-dessus des états signifie $\perp$ plutôt que « encodé », contrairement aux autres chapitres.

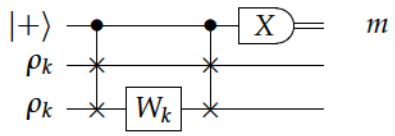


FIGURE 3.3 Circuit de distillation pour la famille d'états $|Y_k\rangle$.

Les portes apparaissant dans ce circuit sont W_k, définie ci-haut, et $\Lambda(\text{Swap})$, où Swap est la porte échangeant deux qubits. Il est implicite que l'application de W_k nécessite l'injection d'un état $|Y_{k-1}\rangle$, préparé au préalable et suffisamment distillé. Comme Swap échange deux qubits, la porte $\Lambda(\text{Swap})$ échange les deux qubits cibles à condition que le qubit de contrôle soit dans l'état $|1\rangle$. Dans le cas qui nous intéresse, le qubit de contrôle étant préparé dans l'état $|+\rangle$, le circuit appliqué jusqu'à la mesure exclusivement a pour effet de superposer « qubits intacts » et « qubits échangés ».

3.5 Analyse des erreurs

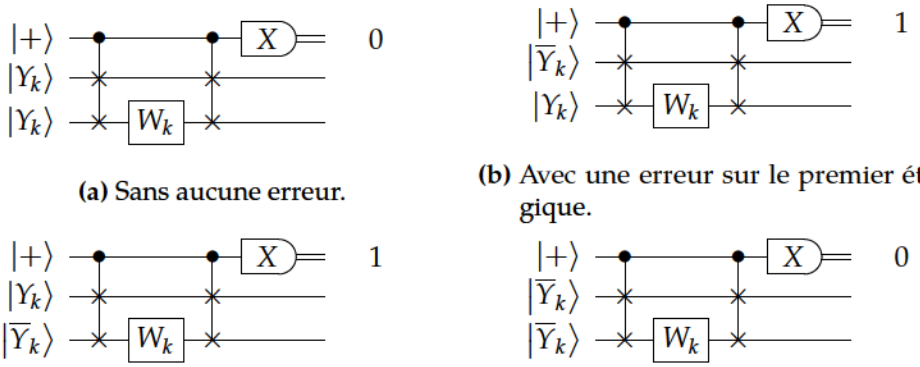
La paire d'états $\{|Y_k\rangle, |\overline{Y}_k\rangle\}$, où $|\overline{Y}_k\rangle$ est l'état orthogonal à $|Y_k\rangle$, définit une base. Un état quelconque peut toujours s'exprimer comme une matrice densité dans cette base :

$$\rho_k = \begin{pmatrix} 1-\epsilon & \Delta \\ \Delta^* & \epsilon \end{pmatrix}. \tag{3.2}$$

Comme ρ_k est une préparation imparfaite de l'état magique $|Y_k\rangle\langle Y_k|$, nous avons $\epsilon, \Delta \ll 1$. Dans la littérature, on suppose typiquement que l'erreur n'est que diagonale, c.-à-d. qu'on pose $\Delta = 0$. Ceci n'est en général pas juste, mais lorsque l'état étudié est $|Y_3\rangle$ ($\theta_3 = \pi/4$), nous pouvons nous en assurer. En effet, appelons W_ψ l'opérateur phase dans la base définie par un état quelconque $|\psi\rangle$:

$$W_\psi |\psi\rangle = |\psi\rangle, \tag{3.3}$$
$$W_\psi |\overline{\psi}\rangle = -|\overline{\psi}\rangle. \tag{3.4}$$

En supposant que nous sachions appliquer W_ψ, nous pouvons toujours éliminer les termes hors-diagonaux de ρ_ψ (Eq. (3.2)) en effectuant la transformation de « tournoiement » (*twirling*)



(a) Sans aucune erreur.

(b) Avec une erreur sur le premier état magique.

(c) Avec une erreur sur le deuxième état magique.

(d) Avec une erreur sur les deux états magiques.

FIGURE 3.4 Résultats de mesure possibles en fonction de différents états d'entrée.

[25] suivante :

$$\rho \to \frac{1}{2}\rho + \frac{1}{2}W_\psi \rho W_\psi^\dagger. \tag{3.5}$$

Or, $W_{Y_3} = W_3 = H$ est une opération de Clifford. Elle peut donc être réalisée efficacement et de manière tolérante aux fautes. En général, nous n'avons pas cette chance. Dans ces travaux, nous ne supposons donc pas que l'erreur est diagonale et nous considérons des états de la forme générale, cf. Eq. (3.2).

Pour se convaincre que le circuit proposé est bien un circuit de distillation, regardons de plus près son effet. La Fig. 3.4 résume les résultats de mesure possibles étant données différentes entrées. Si aucune erreur n'affecte les qubits d'entrée (Fig. 3.4a), alors l'état initial est un état propre +1 de $\Lambda(\text{Swap})$ et de W_k et donc rien ne se produit et la mesure est 0. Si une erreur affecte le premier qubit (Fig. 3.4b) alors l'état évolue de la manière suivante

$$\begin{aligned} \left|+\overline{Y}_k Y_k\right\rangle &\xrightarrow{\Lambda(\text{Swap})} \left|0\overline{Y}_k Y_k\right\rangle + \left|1 Y_k \overline{Y}_k\right\rangle, \\ &\xrightarrow{W_k} \left|0\overline{Y}_k Y_k\right\rangle - \left|1 Y_k \overline{Y}_k\right\rangle, \\ &\xrightarrow{\Lambda(\text{Swap})} \left|-\overline{Y}_k Y_k\right\rangle, \end{aligned} \tag{3.6}$$

et le résultat de mesure est 1. Un raisonnement similaire s'applique si une erreur affecte plutôt le deuxième qubit (Fig. 3.4c). Finalement, si les deux qubits sont erronés (Fig. 3.4d), alors l'état d'entrée est un état propre +1 de $\Lambda(\text{Swap})$ et -1 de W_k. Donc le résultat de mesure est 0, car seulement une phase globale -1 est ajoutée à l'état. Nous distinguons deux cas de figure : si l'état d'entrée est affecté par une seule erreur, la mesure finale donne le

résultat 1, alors que si aucune ou deux erreurs sont présentes, la mesure finale donne 0. En sélectionnant à postériori les instances où 0 est la mesure observée, nous nous assurons que soit aucune, soit deux erreurs se sont produites. En d'autres mots, nous avons supprimé le terme de premier ordre. Évidemment, ceci suppose que les erreurs sur les états d'entrée ne sont pas corrélées. Nous faisons cette supposition.

L'analyse se transpose directement au cas où les entrées sont en fait des matrices densités, ce qui nous permet de couvrir le cas le plus général d'erreurs indépendantes. Dans ce cas, l'état d'entrée ρ_k s'exprime comme le montre l'Eq. (3.7).

$$\rho_k = (1-\delta)|Y_k\rangle\langle Y_k| + \delta|\overline{Y}_k\rangle\langle\overline{Y}_k| + \Delta|Y_k\rangle\langle\overline{Y}_k| + \Delta^*|\overline{Y}_k\rangle\langle Y_k| \tag{3.7}$$

Après l'application du circuit, mais avant la mesure, l'état devient

$$\begin{aligned}|+\rangle\langle+| \otimes \rho_k \otimes \rho_k \rightarrow \frac{1}{2}\Big[&|0\rangle\langle 0|(\rho_k \otimes W_k\rho_k W_k^\dagger) + |1\rangle\langle 1|(W_k\rho_k W_k^\dagger \otimes \rho_k) \\ &+ |0\rangle\langle 1|(\rho_k W_k^\dagger \otimes W_k\rho_k) + |1\rangle\langle 0|(W_k\rho_k \otimes \rho_k W_k^\dagger)\Big]\end{aligned} \tag{3.8}$$

En sélectionnant à postériori le résultat de mesure 0, c.-à-d. en projetant l'état de l'Eq. (3.8) sur l'état $|+\rangle\langle+|_1$ et prenant la trace partielle sur le qubit mesuré, nous obtenons

$$\rho_{\text{out}} \propto \frac{1}{4}\Big[(\rho_k \otimes W_k\rho_k W_k^\dagger) + (W_k\rho_k W_k^\dagger \otimes \rho_k) + (\rho_k W_k^\dagger \otimes W_k\rho_k) + (W_k\rho_k \otimes \rho_k W_k^\dagger)\Big] \tag{3.9}$$

$$\begin{aligned}=&(1-\delta)^2|Y_k\rangle\langle Y_k||Y_k\rangle\langle Y_k| + \delta^2|\overline{Y}_k\rangle\langle\overline{Y}_k||\overline{Y}_k\rangle\langle\overline{Y}_k| \\ &- \Delta^2|Y_k\rangle\langle\overline{Y}_k||Y_k\rangle\langle\overline{Y}_k| - (\Delta^*)^2|\overline{Y}_k\rangle\langle Y_k||\overline{Y}_k\rangle\langle Y_k|\end{aligned} \tag{3.10}$$

En calculant la trace de ρ_{out}, nous obtenons la probabilité de mesurer 0, cf. Eq. (3.11). En prenant plutôt la trace partielle sur le deuxième qubit (l'état est symétrique), nous obtenons l'état de sortie qui sera réutilisé dans la prochaine étape de distillation, cf.Eq. (3.12).

$$\text{Tr}(\rho_{\text{out}}) = 1 - 2\delta + 2\delta^2 \tag{3.11}$$

$$\text{Tr}_2(\rho_{\text{out}}) = (1-\delta)^2|Y_k\rangle\langle Y_k| + \delta^2|\overline{Y}_k\rangle\langle\overline{Y}_k| \tag{3.12}$$

Ces expressions supposent toutefois que l'application du circuit est parfaite. Or, d'une part, ce ne peut être le cas, car $\Lambda(\text{Swap})$ n'est pas une porte de Clifford. Elle sort du cadre de notre supposition initiale qui est que nous puissions réaliser efficacement et de manière robuste les portes de Clifford. Un « gadget » est donc nécessaire : nous utilisons un code stabilisateur à quatre qubits comme intermédiaire. D'autre part, W_k implique l'injection d'un état magique qui est lui-même imparfait. Il faut tenir compte de cette imprécision dans notre calcul d'erreur. L'appendice A de l'article explique comment tenir compte des erreurs

dans l'opérateur phase W_k et dans l'application du $\Lambda(\text{Swap})$ à l'aide du gadget tolérant aux fautes. L'expression complète est beaucoup trop encombrante pour que nous puissions la manipuler. Elle a donc été trouvée et utilisée à l'aide de Mathematica.

3.6 Article

Reducing the quantum computing overhead with complex gate distillation

Guillaume Duclos-Cianci and David Poulin
Département de Physique, Université de Sherbrooke, Québec, Canada
(Dated: April 19, 2015)

In leading fault-tolerant quantum computing schemes, accurate transformations are obtained by a two-stage process. In a first stage, a discrete, universal set of fault-tolerant operations is obtained by error-correcting noisy transformations and distilling resource states. In a second stage, arbitrary transformations are synthesized to desired accuracy by combining elements of this set into a circuit. Here, we present a scheme which merges these two stages into a single one, directly distilling complex transformations. We find that our scheme can reduce the total overhead to realize certain gates by up to a few orders of magnitude. In contrast to other schemes, this efficient gate synthesis does not require computationally intensive compilation algorithms, and a straightforward generalization of our scheme circumvents compilation and synthesis altogether.

I. INTRODUCTION

The accuracy threshold theorem [1–5] states that if a physical device can realize one- and two-qubit operations to an accuracy of approximately 1% [6, 7], then fault-tolerant techniques can be used to reliably quantum-compute with this device for arbitrary long times. This comes at the cost of consuming additional gates and qubits, but in principle this overhead grows 'only' polynomially with the logarithm of the duration of the algorithm. While there are today a few architectures with accuracies near or below threshold, e.g., [8, 9], fault-tolerant quantum computing remains elusive, and a major bottleneck is the prohibitive fault-tolerance overhead. Part of the problem is that the devices' accuracies are too close to the threshold; they should ideally operate one or two orders of magnitude below threshold. But even in such ideal circumstances, the overhead would remain excessively high due to the cost of distillation [10–12] and gate synthesis [13–15].

Because of their continuous nature, it is not possible to error-correct arbitrary quantum operations. Instead, fault-tolerant schemes realize a finite set of discrete, near-perfect universal operations. This universal set of fault-tolerant operations (USFTO) typically includes Clifford operations, since they are naturally fault-tolerant in many encoding schemes, e.g., [1, 7, 16]. Adding any non-Clifford operation to this set renders it universal [17]. Magic state distillation and injection [18] is amongst the most efficient ways to generate these non-Clifford operations.

State injection appends an ancillary register prepared in a magic state to the data register, performs a Clifford transformation on the joint registers, and measures a Pauli operator on the ancillary register. The resulting effect on the data register is a transformation $R(m)$ which depends on the measurement outcome m, cf., Fig. 1 a). Near-perfect magic states are obtained from noisy ones using *state distillation*, a process that uses only Clifford operations. Distilling a magic state to accuracy[1] δ requires a number of noisy input states which grows 'only' polynomially with $\log(1/\delta)$, but even with the best distillation protocols this cost remains substantial [10, 11, 19, 20].

Operations from this USFTO can be assembled to approximately *synthesize* any logical gate to accuracy[2] δ. The cost of synthesizing increases 'only' polynomially with $\log(1/\delta)$ [3, 21], but again for realistic applications, this cost is substantial [13–15]. Moreover, unlike error-correction and distillation overheads, improving the physical devices is of no help; only software improvements can reduce the gate synthesis overhead. This is the problem of gate *compiling*. Only very recently have efficient compilers been discovered [14, 15], and our approach offers an independent and very different solution to the compiling problem.

In this article, we present a scheme to distill a rich family of quantum transformations, which offers several advantages. 1) The total overhead of our scheme can be a few orders of magnitudes lower than what is achieved combining state-of-the art distillation and synthesis techniques. 2) This is achieved by an efficient compilation algorithm. 3) A generalization of our scheme can reduce the gate synthesis cost of any single-qubit gate to a constant.

To get a sense of the overhead associated to distillation and synthesis, suppose that we are using a quantum computer to simulate a molecule with about 100 electronic orbitals. Recent analyses [22–24] show that this requires about 10^{12} non-Clifford rotations, each of a tiny angle $\theta \sim 10^{-7}$, so each need to be accurate to 12 digits to prevent imperfections from building up and scrambling the information. Assuming that Clifford operations can be executed perfectly (thus ignoring the error-correction cost as is usually done in such analysis) and using state-of-the-art compiling sequences [13, 15], this implies that each logical gate requires about 100 operations from the USFTO, for a total of 10^{14} for the entire algorithm. In

[1] Accuracy of distilled states is measured using trace distance.
[2] The accuracy of channels is measured using the operator norm induced by the trace distance.

turn, this implies that magic-state distillation must be accurate to at least 14 digits. Following convention and assuming that noisy magic states can be prepared to accuracy 1%, state-of-the-art distillation protocols [11] require nearly 300 such noisy input states to distill one of sufficient quality. Concluding this example, the total overhead associated to distillation and compilation is over $3 \cdot 10^4$ magic-state inputs on average per logical gate (and even more Clifford operations, which are ignored in our analysis). This represents a major roadblock towards physical realization of fault-tolerant quantum computation.

The rest of this article is organized as follows. In section II, we present the intuition behind our method and highlight key ideas. In section III, we present the distillation circuit and analyze its performance. In the discussion section IV, we revisit the above quantum chemistry simulation problem and propose modifications of our scheme to further reduce the overhead in this setting. We also present a possible improvement of our scheme and discuss future research directions.

II. APPROACH OVERVIEW

To reduce the overhead mentioned in section I, we employ a USFTO which is *over-complete*, in the sense that some operations could be removed from it without affecting its universality. However, removing such redundant gates would affect the synthesis cost. Specifically, our set comprises the Clifford gates (generated by controlled-not $\Lambda(X)$, Hadamard H, and phase gate S), and the infinite family $R_Y(\theta_k) = \exp(-i\pi Y/2^k)$, $k = 3, 4, \ldots$, which are rotations of angle $\theta_k = 2\pi/2^k$ around the y-axis of the Bloch sphere. Note that the cases $k = 1$ and $k = 2$ result in Clifford operations, while $k = 3$ corresponds to the T gate which is commonly used to complete the USFTO. Also, note that in concrete applications where each gate needs only be implemented to a desired accuracy δ, we can effectively truncate the family since for large enough k, the rotation $R_Y(\theta_k)$ can be substituted by the identity to yield an error of magnitude $\delta \approx 2^{-k}$. For this reason we will limit our study to $k < 90$ since larger values have no conceivable utility.

We realize the gates $R_Y(\theta_k)$ by distilling the associated magic states $|Y_k\rangle = \cos(\theta_k/2)|0\rangle + \sin(\theta_k/2)|1\rangle$. These states are injected into the quantum computation using the quantum circuit of Fig. 1 a), which consumes one state $|Y_k\rangle$ and Clifford operations to realize a rotation R_Y by an angle $\pm\theta_k$. The sign of the rotation is completely random but known. This randomness doesn't really impact the synthesis cost as we now explain.

Consider a single qubit rotation $U_{\hat{n}}(\theta)$ of angle θ around axis $\hat{n}$. Existing synthesis schemes can approximate this unitary transformation to absolute accuracy δ at cost $c \log^b(1/\delta)$ where c and b are some constants. As we now demonstrate, the compilation cost using our USFTO scales instead with the relative accuracy $\varepsilon_{\text{rel}} = \delta/|\theta|$ as $3\log(6/\varepsilon_{\text{rel}})/2 + 3$. We decompose the gate using Euler angles as $U_{\hat{n}}(\theta) = R_Z(\alpha)R_Y(\beta)R_X(\gamma) = HS^\dagger R_Y(\gamma)SHR_Y(\beta)S^\dagger R_Y(\alpha)S$, so it requires six Clifford gates and three R_Y rotations of angles $|\alpha|, |\beta|, |\gamma| \leq 2|\theta|$ (see appendix B), each needing to be executed to relative accuracy $\varepsilon_{\text{rel}}/6$. This means that each of these angles can be expressed with $\ell = \log(6/\varepsilon_{\text{rel}})$ significant bits. With $\alpha_k \in \{0, 1\}$, $\alpha = \sum_{k=h}^{h+\ell} \alpha_k 2\pi/2^k$ is a rotation of magnitude 2^{-h} written to relative accuracy $2^{-\ell}$, and is straightforwardly expressed with at most ℓ gates from our USFTO as $R_Y(\alpha) = R_Y(\theta_h)^{\alpha_h} R_Y(\theta_{h+1})^{\alpha_{h+1}} \ldots R_Y(\theta_{h+\ell})^{\alpha_{h+\ell}}$. These ℓ gates are executed sequentially, starting from $k = h + \ell$ down to $k = \ell$. At stage k of this execution, suppose the state injection circuit Fig. 1 produces the outcome -1. The rotation should have been by angle θ_k but this outcome has produced $-\theta_k$ instead, so the state needs to be rotated by angle $2\theta_k = \theta_{k-1}$. This can be fixed by substituting $\alpha \leftarrow \alpha + \theta_{k-1}$, and pursuing the execution of the circuit at $k - 1$. Because of this intrinsic randomness, this execution will require $\ell/2 + 1$ gates on average.

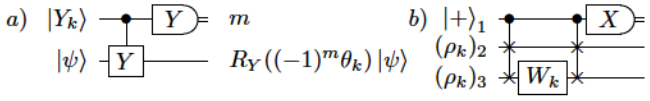


FIG. 1. a) State injection circuit. The controlled qubit is initially prepared in magic state $|Y_k\rangle$ and the target is in an arbitrary state. Following the application of $\Lambda(Y)$, the measurement of the controlled qubit along the y axis with outcome $m = \pm 1$ results in a rotation of $\pm\theta_k$ of the target qubit. b) Distillation circuit for $|Y_k\rangle$. The Λ(SWAP) gates are used to project two noisy versions ρ_k of $|Y_k\rangle$ onto the even-parity subspace, leading to a quadratic improvement of their accuracy.

III. DISTILLATION CIRCUIT AND ANALYSIS

We now explain how to distill the magic states $|Y_k\rangle$. Landahl and Cesare have proposed a distillation protocol for these states that uses a family of shortened Reed-Muller codes [12]. Unfortunately, the Reed-Muller codes are highly inefficient, so any synthesis overhead gained from this approach is overwhelmed by an increased distillation cost at high noise rate, although their approach can offer some advantages if the input magic states are of sufficiently high quality. The overhead of distilling the next simplest non-trivial gate $R_Y(\theta_4)$ ($\sqrt{T}$ gate) using the scheme [12] can be estimated as follows. The distillation requires the use of the 31-qubit quantum Reed-Muller code. There are 1240 undetectable errors of weight 3 in this code, so to leading order, an error rate p is suppressed by $p \to 1240p^3$. The average number of input magic states N used increases as $N \to 31N/P_{\text{acc}}$, where P_{acc} is the probability that no error is detected. Thus, P_{acc} is lower bounded by the probability of one- and two-

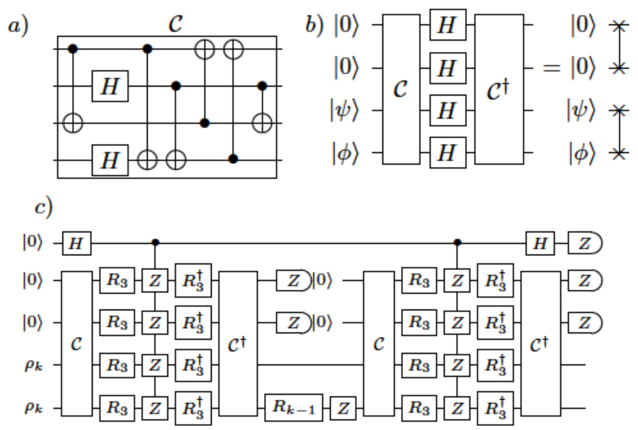


FIG. 2. Detailed implementation of the distillation circuit. a) Encoding circuit for a 4-qubit error-detecting code. b) In this code, the Hadamard gate applied to every qubit has the effect of swapping the two encoded qubits (and the two ancillary qubits). c) Overall circuit combining the primitives of a), b) to implement the circuit of Fig. 1b). We denote $R_k = R_Y(\theta_k)$ for compactness.

qubit errors. Recursing on these relations we find that 1.38×10^5 input magic states of accuracy 1% are required to distill one state of accuracy 3×10^{-15}.

Instead, we build our protocol on the scheme introduced by Meier, Eastin, and Knill [10] and improved by Bravyi and Haah[11] to distill $|Y_3\rangle$. Let us describe their protocol in the more general setting of current interest. The high-level distillation circuit for these states is shown at Fig 1b), detailed implementations are summarized in Fig. 2.

Given the orthogonal basis $|Y_k\rangle$, $|\overline{Y}_k\rangle = Y|Y_k\rangle = i\sin(\theta_k/2)|0\rangle - i\cos(\theta_k/2)|1\rangle$, we define the phase flip operator $W_k = |Y_k\rangle\langle Y_k| - |\overline{Y}_k\rangle\langle\overline{Y}_k|$. A direct calculation shows that $W_k = R_Y(\theta_{k-1})Z$, so the gate W_k can be realized by injecting $|Y_j\rangle$ states with $j < k$. The circuit of Fig. 1.b) performs a measurement of the two-qubit 'parity' operator $M_k = W_k \otimes W_k$. To understand how this leads to error suppression, suppose for simplicity that each input qubit is prepared in the faulty magic state $\sqrt{1-\epsilon}|Y_k\rangle + \sqrt{\epsilon}|\overline{Y}_k\rangle$ (the argument generalizes to arbitrary forms of noise, see åppendix A), such that $\epsilon = 0$ corresponds to a perfect magic state. Their joint state is

$$(1-\epsilon)|Y_k, Y_k\rangle + \epsilon|\overline{Y}_k, \overline{Y}_k\rangle + \sqrt{\epsilon(1-\epsilon)}\left(|Y_k, \overline{Y}_k\rangle + |\overline{Y}_k, Y_k\rangle\right).$$

The first two components of this state are $+1$ eigenstates of $W_k \otimes W_k$ since they have even parity, while the last two have eigenvalue -1 since they have odd parity. Thus, a measurement of $W_k \otimes W_k$ produces result $+1$ with probability $1 - 2\epsilon + 2\epsilon^2$, in which case the post-measurement state will be proportional to $(1-\epsilon)|Y_k, Y_k\rangle + \epsilon|\overline{Y}_k, \overline{Y}_k\rangle$. Since the magnitude of the error component has decreased from $\mathcal{O}(\sqrt{\epsilon})$ to $\mathcal{O}(\epsilon)$, we see that the error has been suppressed quadratically. On the other hand, the result -1 is obtained with complementary probability $2(\epsilon - \epsilon^2)$, in which case the qubits are discarded. This protocol distills a pair of magic states and induces correlated errors, but these correlations can be ignored in our analysis as explained in [11].

An immediate difficulty with this distillation protocol is that the gate $\Lambda(\text{SWAP})$ it uses is not a Clifford transformation. To realize it, we encode a pair of qubits into a 4-qubit *error-detecting* code. The Clifford circuit $\mathcal{C}$ of Fig. 2a) performs the encoding and maps the single-qubit Pauli operators Z_i and X_i as follows:

$$(Z_1, X_1) \to (ZZZZ, XIXX) \quad (1)$$
$$(Z_2, X_2) \to (XXXX, IZII) \quad (2)$$
$$(Z_3, X_3) \to (ZIIZ, XIXI) \quad (3)$$
$$(Z_4, X_4) \to (XIIX, ZIZI). \quad (4)$$

The first two qubits are stabilizer qubits and the last two, logical qubits. A key property seen in this transformation is that exchanging X's for Z's has the effect of swapping the last two lines of the equation, which corresponds to swapping the two encoded qubits. Since the Hadamard gate realizes the X-Z exchange, we deduce that $H^{\otimes 4}$ performs the logical SWAP, cf. Fig. 2b).

We can therefore substitute the $\Lambda(\text{SWAP})$ with four $\Lambda(H)$, but these are still not part of the Clifford group. However, using the identity $H = R_Y(\theta_3)ZR_Y(-\theta_3)$, we can express $\Lambda(H) = \Lambda[R_Y(\theta_3)ZR_Y(-\theta_3)] = R_Y(\theta_3)\Lambda(Z)R_Y(-\theta_3)$. We conclude that $\Lambda(H)$, and therefore $\Lambda(\text{SWAP})$, can be implemented with gates $R_Y(\theta_3)$ from our USFTO and $\Lambda(Z)$, which is a Clifford operation, cf., Fig. 2c).

To recapitulate, the distillation of states $|Y_k\rangle$ requires 1) two noisy input states $|Y_k\rangle$, 2) one near-perfect collection of states $|Y_j\rangle$ for all $j < k$ used to implement the phase inversion gate $W_k = R_Y(\theta_{k-1})Z$, and 3) sixteen near-perfect states $|Y_3\rangle$ used to implement the gates $\Lambda(\text{SWAP})$. This distillation protocol lends itself to a recursive procedure [25] where previously distilled states $|Y_j\rangle$ for $j < k$ are used to distill states $|Y_k\rangle$.

To get the recursion started at $k = 3$, we directly use the protocol of BH [11] to distill $|Y_3\rangle$ states. Using these, we recursively distill $|Y_k\rangle$ for higher values of k. Note that the gates $R_Y(\theta_3)$ are used inside an error-detecting code, cf. Fig. 2c), so they need not be perfect. A -1 measurement outcome at the output of the circuit $\mathcal{C}^\dagger$ reveals that one or more errors occurred in the execution of the encoded SWAP gate. Rejecting the instances where such a non-trivial error syndrome is found suppresses any first order error, thus preserving the quadratic error suppression of the ideal circuit of Fig. 1b).

In general, we can express a noisy magic state ρ_k in the $|Y_k\rangle$, $|\overline{Y}_k\rangle$ basis as

$$\rho_k = \begin{pmatrix} 1-\epsilon_k & \Delta_k \\ \Delta_k^* & \epsilon_k \end{pmatrix} \quad (5)$$

with $0 \leq \epsilon_k \leq 1/2$ and $0 \leq |\Delta_k|^2 \leq \epsilon_k - \epsilon_k^2$, the case $\Delta_k = \epsilon_k = 0$ corresponding to the perfect magic state

$|Y_k\rangle$. For a fixed k and given a set of input noise parameters $(\epsilon_3, \Delta_3, \epsilon_4, \Delta_4, \ldots, \epsilon_k, \Delta_k)$, using computer-assisted calculation we can derive an exact expression for 1) the average output noise parameter (ϵ'_k, Δ'_k) of the distilled state, and 2) the expected number N_j^k of consumed resource states ρ_j of each kind $3 \leq j \leq k$. The expectations are taken over the intrinsic randomness of the protocols, averaging over possible measurement outcomes. This calculation is realized by exactly simulating the quantum circuit of Fig. 2c), which is tractable since it only involves 5 qubits (see appendix A).

Figure 3 shows the overhead, defined as the number of noisy magic input states, required to distill a state $|Y_k\rangle$ to a desired accuracy δ. Consistently with previous studies [10, 11], we assumed that the states $|Y_k\rangle$ can be prepared to accuracy 1%. Note however that for $k > 8$, the state $|0\rangle$ is better than a 1% accurate approximation to $|Y_k\rangle$, and so we substituted all noisy input magic states by $|0\rangle$ for $k > 8$. While we could have performed that substitution for all $k > 3$ — in which case $|Y_3\rangle$ states would have been the only non-Clifford inputs to our protocol — we obtained a slightly lower overhead with this prescription. The green dots on the bottom of the figure which roughly follow the dashed green line labeled BH are the results found in [11]. The dashed green line is a fit of the points assuming that cost $= \mathrm{a}\log^{\mathrm{c}}(1/\delta) + \mathrm{b}$. The green dots above them are all the distillation costs of $|Y_k\rangle$ for various numbers of rounds of distillation and for $4 \leq k \leq 45$. They were recursively obtained from the discrete dots of BH using the rules of thumb described in appendix A. Remember that after $k = 8$, the initial states are $|0\rangle$, which are free and are more and more accurate, as k increases. Moreover, their "error" is purely off-diagonal since they are pure states. Combined with our suboptimal rules of thumb, this explains why the dots seem to behave non monotonically. A sample of the data is given in Appendix C. The dark diamonds are the costs of implementing the θ_4 rotation using $|Y_4\rangle$ and $|Y_3\rangle$ states distilled to various precisions. The blue line running through is a fit of these points. The dark squares are the costs of implementing the smallest rotation such that $\theta_k < 8\delta$ for various precisions, using states distilled once, twice or three times. The blue line running through is also a fit. The red curves were derived using fits, they therefore provide an unrealistic advantage to the corresponding schemes since they ignore the discreteness of the accessible rotations. First, we used the fit of the various protocols to obtain the T-count for a given accuracy δ. Then, we used the fit of BH dots to obtain the cost of T rotations of accuracy δ/T-count. Multiplying both gives the red lines.

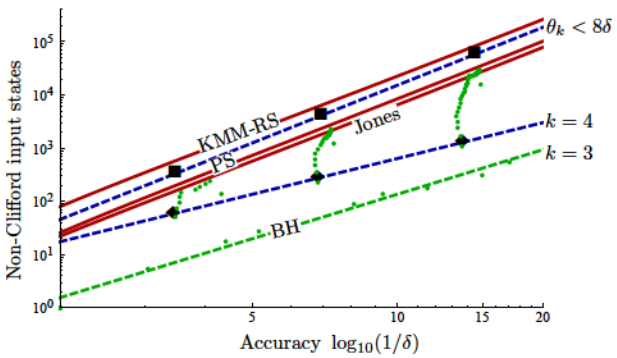


FIG. 3. (Color online) Overhead, measured as the number of input non-Clifford states, to realize a gate to accuracy δ. The diamonds are the overhead to realize $R_Y(\theta_4)$. Green dots are distillation costs of $|Y_k\rangle$, with $4 \leq k \leq \log_2(2\pi/\delta)$ (smaller angles $\theta_k < \delta$ can be substituted by 0 to accuracy δ.) Squares represent the cost of an arbitrary rotation of angle $\theta \leq 8\delta$. Red lines are obtained by combining the distillation scheme BH of [11] with the gate synthesis KMM-RS of either [13] or [15] (both give very similar overhead), or PS of [26], or Jones of [20]. Note that in the latter case, the resource state is a register of size $\log \delta$.

IV. DISCUSSION

A. Quantum chemistry simulation

Returning to our molecule simulation example with a $\delta = 10^{-12}$ target accuracy, we see on Fig. 3 (KMM-RS line) the 10^4 overhead obtained by combining the distillation scheme [11] and synthesis scheme [13]. This overhead is to realize the target gate $R_Z(1/10)$, but we note that these protocols are largely insensitive to the target gate. In contrast, the overhead of our protocol depends on the target gate, and it ranges between 100-10,000 (green dots on Fig. 3) for the family of gates $R_Y(\theta_k)$. While this improvement is realized for specific single qubit gates $R_Y(\theta_k)$, for arbitrary rotations, the overhead shown on Fig. 3 increases only with the logarithm of the relative accuracy of the rotation ($\varepsilon_{\mathrm{rel}} \sim 10^{-5}$ for molecule simulation), as explained above.

We now explain how to modify the quantum simulation algorithm to circumvent this logarithmic compilation overhead altogether. The simulation of quantum mechanical systems on quantum computers is based on the Trotter-Suzuki decomposition

$$e^{A+B} = \lim_{n\to\infty} (e^{A/n} e^{B/n})^n$$

which, for finite n, gives an approximation with error $\mathcal{O}(1/n)$. For a many-body Hamiltonian relevant for the simulation of molecules $H = \sum_a h_a$, the time evolution operator $U = e^{-iHt}$ can therefore be approximated by

$$U = V^n = \left[e^{-ih_1 t/n} e^{-ih_2 t/n} \ldots\right]^n \qquad (6)$$

where each $e^{-ih_a t/n}$ is (up to Clifford transformations) a rotation $R_Y(\theta_a)$ by some angle θ_a of magnitude roughly $10^{-7} \lesssim 2\pi/2^{20}$. Moreover, each of these rotations needs to be implemented to accuracy 10^{-12}, so each θ_a can be written in binary with about 16 significant bits. The idea of coalescing is to replace the n identical repetitions of the sequence V as in Eq. (6) by a repetition of n non-identical sequences $V_1 V_2, \dots V_n$. Each sequence V_j contains (up to Clifford transformations) only rotations $R_Y(\theta_{20})$. The rotation associated to the term h_a appears in each sequence V_j with probability $\theta_a/(2\pi/2^{20})$, in such a way that the average rotation per sequence is precisely θ_a. This shows that the gates $R_Y(\theta_k)$, which have no compilation cost in our scheme, can naturally occur in quantum simulations. More broadly, they occur very naturally in many quantum algorithms; for instance they are the only non-Clifford gates appearing in the quantum Fourier transform circuit [27].

Lastly, for arbitrary rotations, our scheme does slightly worst than existing schemes. We note however that it can be generalized straightforwardly to the distillation of the family of states $|Y_k^m\rangle = |Y(m2\pi/2^k)\rangle$ for a fixed integer $0 < m < 2^k$: distillation of $|Y_k^m\rangle$ can be realized given access to distilled $|Y_j^m\rangle$ with $j = 3, 4, \dots k-1$. Since any rotation can be written as $m2\pi/2^k$ to k bits of accuracy, this provides a way of realizing any rotation $R_Y(\theta)$ using on average only two distilled magic states. This approach entirely avoids the need to compile (aside from an Euler angle decomposition), has a constant gate synthesis cost, and pushes all the gate synthesis overhead into the distillation. This last feature is important since distillation occurs off-line, i.e. it does not involve the data qubits. Similar protocols were introduced in [29] and [30].

B. Further improvements

In this section we discuss possible ways of further reducing the compilation/distillation overhead of our scheme. From the start, we note that our protocol is equivalent to [11] when distilling the T gate (i.e. $k = 3$), and that this forms the base of our recursion. The vast majority of studies on magic state distillation has focused on the T gate, and any future improvement there can be directly incorporated into our protocol by substituting it for the first step of our recursion.

One clear path to improvements is to use the same distillation tools with a thoroughly optimized distillation schedule (see appendix A). In our numerical analysis, we have used a rule of thumb which consists in setting the contribution to the final error from every noise source to be equal, ignoring the fact that different components have different costs. Accounting for these costs would lead to a reduced overhead: the schedule should permit a larger contribution to the final error from a costly component.

The central component of the distillation scheme is the controlled-SWAP gate. Following [10], it is implemented inside a 4-qubit quantum code where it can be substituted by 4 $\Lambda(H)$, and each of those can further be substituted by two $|Y_3\rangle$ state injections. Since one distillation round requires two Λ(SWAP) and distills two qubits, we obtain a rate of 1/8 distilled qubit per $|Y_k\rangle$ states consumed. Increasing this rate would reduce the overhead.

For the distillation of states $|Y_k\rangle$ with $k > 3$, a rate 1/7 can be obtained by realizing the Λ(SWAP) directly on the data, not making use of any code. Indeed, the Λ(SWAP) can be realized with seven T gates [28]. However, by doing so we would loose the benefit of the additional noise suppression offered by the code, so it is not obvious that this would be beneficial, at least early in the distillation schedule when the noise is relatively high.

It is possible to replace the 4-qubit code with a different code to achieve a rate $m/4(m+1)$, where m is any positive integer. This is asymptotically a two-fold improvement of the yield, and given the recursive nature of our protocol this gain can be amplified to a more substantial value. Specifically, the code family has parameters $[[2m+2, 2m, 2]]$; it is described by the stabilizer generators $Z^{\otimes 2m+2}$ and $X^{\otimes 2m+2}$ and has logical operators

$$\overline{Z}_j = \prod_{i=0}^{2j+1} Z_i, \quad \overline{X}_j = X_{2j+1} X_{2j+2} \quad \text{for } 0 \le j \le m-1 \tag{7}$$

$$\overline{Z}_j = \prod_{i=0}^{2j+1} X_i, \quad \overline{X}_j = Z_{2j+1} Z_{2j+2} \quad \text{for } m \le j \le 2m-1. \tag{8}$$

The 4-qubit code used above corresponds to the special case $m = 1$. Just like the 4-qubit code, this code has the property that swapping all X and Z operators, which is realized with the transversal Hadamard gate, has the effect of swapping pairs of logical qubits j with $j+m$. Thus, this enables an m-fold parallelization of our original scheme at higher encoding rate. Note however that this code still detects only a single error, and that by increasing m we create more opportunities for errors to accumulate. As a consequence, if each $|Y_3\rangle$ state used to implement the $\Lambda(H)$ is accurate to ϵ, the probability of detecting an error scales like $m\epsilon$, and the probability of a harmful undetected error scales like $m^2\epsilon^2$. While this is a deterioration over the case $m = 1$, it offers an additional flexibility in the distillation schedule that can be greatly beneficial, as we now explain.

As can be seen on the BH data of Fig. 3, only very sparse values of the accuracy δ can be realized with standard distillation protocols. This is because the error is essentially squared at each iteration with a fixed prefactor, i.e., $\epsilon \to c\epsilon^2$, leading to the discrete set of values $\epsilon, c\epsilon^2, c(c\epsilon^2)^2$, etc. This coarseness has the drawback that we will sometimes be forced to use un-necessarily accurate and costly gates, simply because there is a large gap in the range of available accuracies. This problem occurs not only during the implementation of the algorithm, but

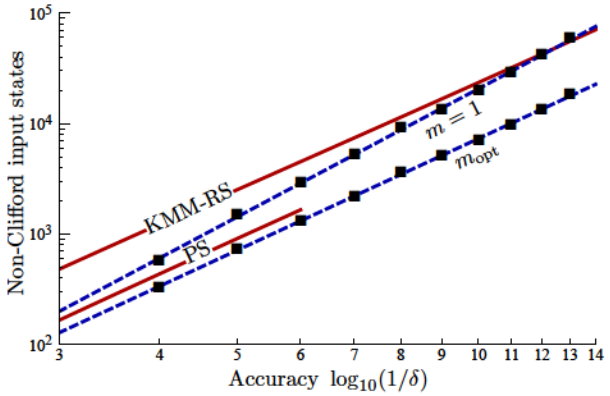


FIG. 4. (Color online) As in Fig. 3. Approximate overhead of a rotation $R_Y(\theta)$ by angle $\theta < 8\delta$ (relative accuracy 1/8). Compares the original protocol $m = 1$ with a protocol allowed to use higher rate codes $m_{\rm opt}$. For a precision of $\epsilon = 10^{-13}$, a factor improvement of ~ 3.2 is observed.

in the distillation procedure itself where previously distilled states $|Y_j\rangle$ are used to assist the distillation of $|Y_k\rangle$ with $k > j$. With the enlarged code family proposed here, we can use the parameter m to fine tune the accuracy of the distilled states. Combined with the improved rate, this has the potential to lead to substantial savings. Fig. 4 shows gains obtained by choosing the optimal value of m at each step of the distillation, and demonstrates up to 3-fold improvement over the case $m = 1$.

We note that this calculation was realized using the leading order expansion described in the appendix A, adapted to the case $m > 1$. Moreover, during the distillation of $|Y_k\rangle$, we assumed that states $|Y_j\rangle$ with $j < k$ of arbitrary accuracy ϵ_j could be accessed at a cost $C_j(\epsilon)$, where this cost function was obtained by fitting a discrete set of achieved accuracies. In other words, we ignored the coarseness of the achievable accuracies. Since one of the main advantage of the schemes with $m > 1$ is the possibility to finely tune the accuracy, we expect a more complete calculation to yield an even larger improvement over the $m = 1$ case.

V. CONCLUSION

In conclusion, we have presented a scheme to distill complex magic states which can offer significant savings over the traditional distillation/synthesis approaches. There are many foreseeable ways to obtain further gains from our scheme. We have investigated a generalization of our scheme based on a family of high rate codes that achieve a $m/4(m+1)$ yield for integer m, resulting in an additional 3-fold reduction of the overhead. Further savings could be obtained using the approach of [26] to find more efficient distillation circuits. Also, since the improvements we obtain are particularly drastic for the specific gates $R_Y(\theta_k)$, we could generalize the algorithm of [15] to optimize compiling sequences for Clifford and these gates.

Acknowledgements— This work was supported by Canada's NSERC and Québec's FRQNT through the network INTRIQ. DP acknowledges the hospitality of The University of Sydney where this work was completed.

Appendix A: Error Analysis

In this appendix we give the details of the calculations and simulations for the circuits of Fig. 2c). In general, we can express a noisy magic state ρ_k in the $|Y_k\rangle$, $|\overline{Y}_k\rangle$ basis as

$$\rho_k = \begin{pmatrix} 1-\epsilon_k & \Delta_k \\ \Delta_k^* & \epsilon_k \end{pmatrix}. \tag{A1}$$

For a fixed k and given a set of input noise parameters $(\epsilon_3, \Delta_3, \epsilon_4, \Delta_k, \ldots, \epsilon_k, \Delta_k)$, using computer-assisted calculation we can derive an exact expression for 1) the average output noise parameter (ϵ'_k, Δ'_k) of the distilled state, and 2) the expected number N_j^k of consumed resource states ρ_j of each kind $3 \le j \le k$. The expectations are taken over the intrinsic randomness of the protocols, averaging over possible measurement outcomes.

To leading order, we expect the following scaling for the diagonal noise parameter

$$\epsilon_k^{\rm out} \approx \epsilon_k^2 + 2\binom{8}{2}\epsilon_3^2 + \epsilon_{k-1} + \frac{1}{2}\epsilon_{k-2} + \frac{1}{4}\epsilon_{k-3} \ldots + \frac{1}{2^{k-4}}\epsilon_3,$$

where it is implicitly assumed that $\epsilon_j = 0$ for $j < 3$ (Clifford operations). The first term comes from the ideal distillation circuit of Fig. 1b) which quadratically reduces the error. The second term comes from the 8 $R_Y(\theta_3)$ gates used to implement the Λ(SWAP) inside the error-detecting code. It takes 2 faults out of these 8 gates to produce an undetected error. The extra factor of 2 accounts for the two occurrences of the Λ(SWAP) in the protocol. Finally, the last terms come from the $W_k = R_Y(\theta_{k-1})$ gate which consumes one $|Y_{k-1}\rangle$ state, consumes one $|Y_{k-2}\rangle$ state with probability 1/2, etc. We note that in general, we can use $|Y_3\rangle$ states of different accuracies to implement the W_k and the Λ(SWAP), since the latter appears inside a code, but we will omit this detail here for simplicity.

Similarly, we can estimate the expected number N_j^k of states ρ_j consumed during one distillation round of $|Y_k\rangle$ to be $N_j^k = [2^{j-k+1} + 16\delta_{j,3}]\bar{r}$, where $\delta_{j,3}$ is the Kronecker delta and $\bar{r} \approx (1 + 16\epsilon_3 + 2\epsilon_k)$ is the average number of times the protocol needs to be repeated before all five measurement outcomes in Fig. 2c) return the value $+1$. A -1 outcome can be obtained either when one of the 16 $R_Y(\theta_3)$ gate is faulty or when one of the two input $|Y_k\rangle$ states are faulty. The behaviour and dependence of the off-diagonal terms Δ_k is more difficult to derive intuitively, but we note that their value had essentially no effect on the exact calculation; setting all $\Delta_j = 0$ had no significant impact on our results.

To complete the analysis, we need a distillation schedule. To distill a state $|Y_k\rangle$ to accuracy δ, our scheme makes use of previously distilled $|Y_j\rangle$ states for $j < k$ with given noise parameters (ϵ_j, Δ_j). To what accuracy should these resource states have been previously distilled? If they were not sufficiently distilled, their use in the distillation of $|Y_k\rangle$ could actually increase its error. On the other hand, using states $|Y_j\rangle$ that were distilled to a much greater precision than the targeted accuracy δ is wasteful. While we have not thoroughly optimized the distillation schedule, we used the following rule of thumb. A perfect distillation circuit gives $\epsilon_k^{\text{out}} = (\epsilon_k^{\text{in}})^2$. With the use of imperfect magic states $|Y_j\rangle$, this output accuracy is instead $\epsilon_k^{\text{out}} \approx (\epsilon_k^{\text{in}})^2 + \sum_{j<k} \alpha_j \epsilon_j^{\beta_j}$ for some integers α_j and β_j. Given this, we use resource states $|Y_j\rangle$ of accuracy $\epsilon_j \approx [(\epsilon_k^{\text{in}})^2/\alpha_j]^{1/\beta_j}$. The intuition behind this rule is that each source of error will contribute equally to the output error ϵ_k^{out}.

In practice, we had one rule for the $|Y_3\rangle$ states involved in the $\Lambda(\text{SWAP})$ and another rule for the states involved in W_k. For the first one, we chose $|Y_3\rangle$ states with biggest error such that it is still smaller than $\epsilon_k^{\text{in}}/4$, i.e. $\epsilon_3 < \epsilon_k^{\text{in}}/4$. Given that four copies are involved, the first order error rate is $4 \times \epsilon_k^{\text{in}}/4$ and since first order errors are suppressed, the contribution to the output error is $\mathcal{O}\big((\epsilon_k^{\text{in}})^2\big)$. For the second rule, we simply took $|Y_k\rangle$ states with biggest error such that still $\epsilon_j^2 < 10(\epsilon_k^{\text{in}})^2$. The factor of 10 was found by trial and error, the problem being that without it, if $\epsilon_j \gtrsim \epsilon_k^{\text{in}}$, then the protocol would take the $|Y_j\rangle$ distilled one more round such that in the end $\epsilon_j \sim (\epsilon_k^{\text{in}})^2$, which is an overkill. With this choice, the contribution to the output error is still $\mathcal{O}\big((\epsilon_k^{\text{in}})^2\big)$.

In the following sections, we give a more detailed account of the error analysis.

1. Imperfect W_k

Performing the phase flip operator W_k requires the use of resource states $|Y_j\rangle$ $(0 \leq j < k)$ which are imperfect. First, the state $|Y_{k-1}\rangle$ is injected using the circuit of Fig. 1a). In order to account for errors in the magic state, we write down its effect (we label the top wire 1 and the bottom wire, 2)

$$\rho_{k-1} \otimes \rho \to |\pm i\rangle\langle\pm i|_1 \Lambda(Y_2)(\rho_{k-1} \otimes \rho)\Lambda(Y_2)|\pm i\rangle\langle\pm i|_1 \tag{A2}$$

where we drop normalization and where the $\pm$ sign is determined by the measurement outcome. If ρ_{k-1} is perfect, then the circuit applies $R_Y(\pm\theta_{k-1})$ to ρ_k. Otherwise, recall that $|\overline{Y}_{k-1}\rangle = Y|Y_{k-1}\rangle$. Since, $Y = iXZ$ and that a) Z_1 commutes with any controlled unitary $\Lambda(U_2)$, and that b) X_1 propagates to $X_1 \otimes U_2$ if U_2 is self-adjoint, then $Y_1 \mathbb{1}_2$ propagates to $Y_1 Y_2$ through $\Lambda(Y_2)$. Tracing over the resource state, the effect of injecting an imperfect state is

$$\rho \xrightarrow{m=\pm 1} \tag{A3}$$
$$(1-\epsilon_{k-1})\rho_\pm + \epsilon_{k-1} Y\rho_\pm Y \pm \Delta_{k-1} Y\rho_\pm \pm \Delta_{k-1}^* \rho_\pm Y,$$

where we take advantage of the fact that $[R_Y(\theta), Y] = 0$ and where we have defined $\rho_+ = R_Y(\theta_{k-1})\rho R_Y^\dagger(\theta_{k-1})$ corresponding to measurement outcome $+1$ and $\rho_- = R_Y(-\theta_{k-1})\rho R_Y^\dagger(-\theta_{k-1})$ corresponding to measurement outcome -1. The states ρ_+ and ρ_- are obtained with equal probabilities. When ρ_- is obtained, a $R_Y(\theta_{k-2})$ correction is required which involves another faulty resource state. Errors propagate again and combine with previous ones. However, errors are always of Y-type. For example, if we inject another rotation, in order to have no overall error, the same error has to happen on both injections. Similarly a $Y\rho$ error in the first injection and a ρY error in the second one results in a $Y\rho Y$ error, etc. We define the error amplitude vector $\vec{\epsilon}_k = (1-\epsilon_k, \epsilon_k, \Delta_k, \Delta_k^*)$ for imperfect resource state ρ_k. These combinations of errors define a product, noted $\times$, on such vectors:

$$\begin{aligned}&(a,b,c,d) \times (e,f,g,h) \stackrel{\text{def}}{=} \\ &(ae+bf+cg+dh, af+be+ch+dg, \\ &ag+bh+ce+df, ah+bg+cf+de).\end{aligned} \tag{A4}$$

Using transpositions, τ_{ij}, this can be rewritten

$$\begin{aligned}&A \times B \\ &= (A\cdot B, A\cdot\tau_{12}\tau_{34}B, A\cdot\tau_{13}\tau_{24}B, A\cdot\tau_{14}\tau_{23}B),\end{aligned} \tag{A5}$$

where "$\cdot$" is the usual scalar product of vectors.

We note $\vec{\epsilon}_{k-} = (1-\epsilon_k, \epsilon_k, -\Delta, -\Delta^*)$, the error applied when the measurement outcome is -1. Using the vector product just defined, the imperfect application of $R_Y(\theta_{k-1})$ gives an expression for $\vec{\epsilon}_{W_k}$,

$$\vec{\epsilon}_{W_k} = \frac{1}{2^{k-3}}\Big(\prod_{j=3}^{k-1}\vec{\epsilon}_{j-}\Big) + \sum_{i=3}^{k-1}\frac{1}{2^{k-i}}\Big(\prod_{j=i+1}^{k-1}\vec{\epsilon}_{j-}\Big) \times \vec{\epsilon}_i. \tag{A6}$$

The first term corresponds to having to apply all rotations $k > j \geq 3$. This occurs with probability $1/2^{k-3}$. The second term sums over all other possibilities: for a given value i, we assume injections $k > j > i$ have measurement outcome -1 and that the ith measurement outcome is $+1$. This happens with probability $1/2^{k-i}$.

Using the resulting error vector $\vec{\epsilon}_{W_k}$, the imperfect phase operator as the following effect (omitting normalization)

$$\begin{aligned}\tilde{W}_k(\rho) \to &(\vec{\epsilon}_{W_k})_1 W_k\rho W_k + (\vec{\epsilon}_{W_k})_2 Y W_k\rho W_k Y \\ &+ (\vec{\epsilon}_{W_k})_3 Y W_k\rho W_k + (\vec{\epsilon}_{W_k})_4 W_k\rho W_k Y.\end{aligned} \tag{A7}$$

2. Λ(Swap) gadget

The circuit of Figure 2.c) shows that a $\Lambda(\text{Swap})$ gadget uses eight $R_Y(\theta_3)$ rotations, each requiring a $|Y_3\rangle$

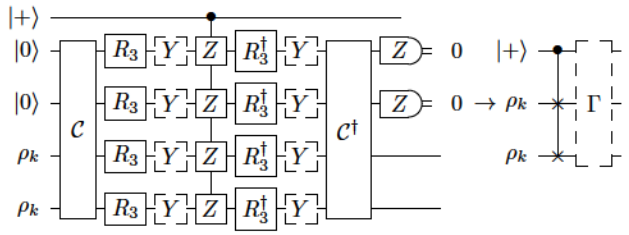


FIG. 5. Passing the noise through the gadget. The dashed boxes represent the resulting effective stochastic noise maps.

injection. As explained in appendix B.1, errors in the resource state translate into errors on the target qubit. However, since $W_3 = H$ (Clifford), we can perfectly twirl all $|Y_3\rangle$ states such that the error is always diagonal: $\rho_3 = (1-\gamma)|Y_3\rangle\langle Y_3| + \gamma|\overline{Y}_3\rangle\langle\overline{Y}_3|$. Consequently, we account for errors in the $\Lambda(\text{Swap})$ gadget by following each R_{Y_3} rotations by a Y-Pauli channel of strength γ: $\rho \to (1-\gamma)\rho + \gamma Y\rho Y$.

To keep the simulation on only three qubits, we push these errors through the gadget and project on the trivial syndrome, as is shown in Fig. 5. In the following, we label wires 0 to 4 from top to bottom. To propagate Y_i errors through a $\Lambda_0(H_i)$, we note that $\{Y_i, H_i\} = 0$ implies $Y_i\Lambda_0(H_i) = \Lambda_0(H_i)Z_0Y_i$. Then, we can propagate Y_i errors through the decoding circuit $\mathcal{C}^\dagger$, using Eqs. (1-4). By determining the effect of every single-qubit Y error this way, we can deduce the effective stochastic map Γ, see Fig. 5. Note that the way it is now defined, Γ is not trace preserving. We define $Pr^{(1,2)}_{\Lambda(\text{Swap})} = \text{Tr}(\Gamma(\rho^{(1,2)}))$, the probability of measuring the trivial syndrome in the first or second $\Lambda(\text{Swap})$ gadget.

3. Error Calculation and Simulation Details

Using the results of the two previous subsections, we calculate the single-qubit output state as a function of the various inputs in the following manner:

1. Set $\rho = |+\rangle\langle+| \otimes \rho_k \otimes \overset{1}{\rho_k}$
2. Set $\rho^{(1)} = \Gamma(\Lambda(\text{Swap})\rho\Lambda(\text{Swap})^\dagger)$
3. Compute $\text{Pr}^{(1)} = \text{Tr}(\rho^{(1)})$
4. Normalize $\rho^{(1)} \leftarrow \rho^{(1)}/\text{Pr}^{(1)}$
5. Set $\rho^{(W_k)} = \tilde{W}_k(\rho^{(1)})$
6. Set $\rho^{(2)} = \Gamma(\Lambda(\text{Swap})\rho^{(W_k)}\Lambda(\text{Swap})^\dagger)$
7. Compute $\text{Pr}^{(2)} = \text{Tr}(\rho^{(2)})$
8. Normalize $\rho^{(2)} \leftarrow \rho^{(2)}/\text{Pr}^{(2)}$
9. Post-select on measuring the trivial syndrome: $\rho^{\text{out}} = \text{Tr}_{1,2}(|+\rangle\langle+| \otimes \mathbb{1} \otimes \mathbb{1}\rho^{(2)})$
10. Compute $\text{Pr}^{\text{out}} = Tr(\rho^{\text{out}})$

From normalized ρ^{out}, we can extract $\epsilon^{\text{out}} = \epsilon^{\text{out}}(\gamma, \epsilon_k, \delta_k)$ and $\Delta^{\text{out}} = \Delta^{\text{out}}(\gamma, \epsilon_k, \delta_k)$. Using these quantities, we define the cost of a distillation round:

$$\begin{aligned}\text{Cost}_k(\epsilon^{\text{out}}, \Delta^{\text{out}}) = \\ \frac{1}{2}\Bigg(\frac{2\text{Cost}_k(\epsilon_k) + 8\text{Cost}_3(\epsilon_{3,\Lambda(\text{Swap})})}{\text{Pr}^{(1)}} \\ + \text{Cost}_{W_k}(\{\epsilon_j, \Delta_j\}) \\ + 8\text{Cost}_3(\epsilon_{3,\Lambda(\text{Swap})})\Bigg)\frac{1}{\text{Pr}^{(2)}\text{Pr}^{\text{out}}}\end{aligned} \tag{A8}$$

where $3 \le j < k$. The $1/2$ prefactor accounts for the fact that we distill two copies of ρ_k. The cost to apply a W_k operator is the sum of the cost of all states $3 \le j < k$ multiplied by the probability of being needed:

$$\text{Cost}_{W_k}(\{\epsilon_j, \Delta_j\}) = \sum_{j=3}^{k-1} \frac{\text{Cost}_j(\epsilon_j, \Delta_j)}{2^{k-1-j}}. \tag{A9}$$

For $3 \le k \le 8$, we initialize the cost to $\text{Cost}_k(0.01, 0) = 1$ and the state to $\rho_k = 0.99|Y_k\rangle\langle Y_k| + 0.01|\overline{Y}_k\rangle\langle\overline{Y}_k|$. For $k \ge 9$, the state $|0\rangle$ is actually closer in trace distance to the resource state than an imperfectly prepared version with 1% error. In this case, we initialize $\rho_k = |0\rangle\langle 0|$ and $\text{Cost}_k(\epsilon_{k,0}, \Delta_{k,0}) = 0$, where $\epsilon_{k,0}$ and $\Delta_{k,0}$ are obtained by expressing $|0\rangle\langle 0|$ in the basis defined by $|Y_k\rangle$. We find $\epsilon_{k,0} = \sin^2(\theta_k/2)$ and $\Delta_{k,0} = \sin(\theta_k)/2$. In this case, the error is purely off-diagonal, i.e. it saturates the bound $|\Delta|^2 \le \epsilon - \epsilon^2$ derived from the positivity of ρ_k.

For $k = 3$, the cost table is given by the scheme of [11]. Starting at $k = 4$, two copies $\text{Cost}_4(0.01, 0) = 1$ are used to distill two improved copies. This gives a new value $\text{Cost}_k(\epsilon_{k,1}, \Delta_{k,1}) = \text{Cost}_k(\epsilon^{\text{out}}, \Delta^{\text{out}})$. Using two copies of this improved state, another round of distillation is performed and so on. Remember that the precisions and costs of the inputs are chosen according to the rule of thumbs discussed earlier. Once a set of values has been obtained for the distillation of $|Y_k\rangle$, $|Y_{k+1}\rangle$ can be distilled and so on, bootstrapping on the protocol of [11].

4. Errors measures

In this work, errors on magic states were measured using the trace distance and gate errors resulting from the injection of these states were measured using the operator norm induced by the trace distance. Both are 1-norms. In other papers, the error measure for states is also the trace distance. However, the error measure for compilation sequences is a 2-norm on unitaries. In order to compare these different measures, we note that the 2-norm upper bounds the 1-norm by a constant factor $\sqrt{2}$. We took this factor into account in the figures, but note that it is negligible on the log log scale of the x-axis of Fig. 3.

Appendix B: Euler angle decomposition

In this appendix, we demonstrate the claim made in the main text that a small-angle single-qubit rotation $R_{\hat{n}}(\theta)$ can be decomposed into a sequence of rotations $R_Z(\alpha)R_Y(\beta)R_X(\gamma)$ around the three axes of the Bloch sphere, all with angles of magnitude bounded by 2θ. On one hand, in the axis-angle representation, the rotation matrix takes the form

$$R_{\hat{n}}(\theta) = \begin{pmatrix} \cos\theta + n_x^2(1-\cos\theta) & n_x n_y(1-\cos\theta) - n_z\sin\theta & n_x n_z(1-\cos\theta) + n_y\sin\theta \\ n_y n_x(1-\cos\theta) + n_z\sin\theta & \cos\theta + n_y^2(1-\cos\theta) & n_y n_z(1-\cos\theta) - n_x\sin\theta \\ n_z n_x(1-\cos\theta) - n_y\sin\theta & n_z n_y(1-\cos\theta) + n_x\sin\theta & \cos\theta + n_z^2(1-\cos\theta) \end{pmatrix}. \tag{B1}$$

On the other hand, in the Euler angle decomposition, this rotation matrix is

$$R_Z(\alpha)R_Y(\beta)R_X(\gamma) = \begin{pmatrix} \cos\beta\cos\alpha & \cos\gamma\sin\alpha + \sin\gamma\sin\beta\cos\alpha & \sin\gamma\sin\alpha - \cos\gamma\sin\beta\cos\alpha \\ -\cos\beta\sin\alpha & \cos\gamma\cos\alpha - \sin\gamma\sin\beta\sin\alpha & \sin\gamma\cos\alpha + \cos\gamma\sin\beta\sin\alpha \\ \sin\beta & -\sin\gamma\cos\beta & \cos\gamma\cos\beta \end{pmatrix}. \tag{B2}$$

By changing orientation of $\hat{n}$, we can assume without loss of generality that $0 \le \theta \le \pi/4$. Equating these two matrices and noting that $|n_x|,\ |n_y|,\ |n_z| \le 1$, element (3,1) yields the inequality

$$|\sin\beta| \le 1 - \cos\theta + \sin\theta \le \sin 2\theta, \tag{B3}$$

which implies that $|\beta| \le 2\theta$. We proceed similarly for the other angles. Equating element (2,1) of these matrices yields

$$\begin{aligned} |\sin\alpha| &\le (1 - \cos\theta + \sin\theta)/|\cos\beta| && \text{(B4)} \\ &\le (1 - \cos\theta + \sin\theta)/\cos 2\theta \le \sin 2\theta, && \text{(B5)} \end{aligned}$$

which implies that $|\alpha| \le 2\theta$. Finally, the bound on γ is obtained following an identical reasoning using element $(3,2)$ of the matrix equality.

Appendix C: Data

In this appendix, we give a sample of the data resulting from our distillation protocol of $|Y_k\rangle$ states. $|Y_3\rangle$ states are given by the protocol of BH [11], so they are not listed below. Table I lists pairs of $\Big(\log_{10}(1/\delta), \text{Cost}\Big)$ as a function of k, labeling the states, and the number of rounds of distillation. The accuracy δ is the trace distance between the resource state and $|Y_k\rangle$. Note that for $k \ge 9$, $|0\rangle$ is the initial resource state. It is pure and in this sense, we say that its error is purely off-diagonal.

Table II lists a sample of the resources used to perform distillation. The states are labeled by two parameters: k the level of the state in the family and "#rounds", the number of times it has been distilled. Thus, a (6,3) state is a $|Y_6\rangle$ state distilled 3 times. The first column lists a sample of the states to be distilled. The second column gives which states $|Y_3\rangle$ are used in the Λ(Swap) gadgets. The third column gives the states that should be used to perform the W_k operation. These numbers were all obtained using our "rules of thumb".

[1] P. W. Shor, in *Proceedings of the 37th Symposium on the Foundations of Computer Science* (IEEE press, Los Alamitos, California, 1996) pp. 56–65, quant-ph/9605011.

[2] D. Aharonov and M. Ben-Or, in *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997) quant-ph/9611025.

[3] A. Y. Kitaev, Russ. Math. Surv., **52**, 1191 (1997).

[4] E. Knill, R. Laflamme, and W. H. Zurek, Phil. Trans. R. Soc. Lond. A., **454**, 365 (1998), quant-ph/9702058.

[5] J. Preskill, Proc. R. Soc. Lond. A, **454**, 385 (1998), quant-ph/9705031.

[6] E. Knill, Nature, **434**, 39 (2005), quant-ph/0410199.

[7] R. Raussendorf and J. Harrington, Phys. Rev. Lett., **98**, 190504 (2007).

[8] M. Devoret and R. Schoelkopf, Science, **339**, 1169 (2013).

[9] P. Schindler, D. Nigg, T. Monz, J. T. Barreiro, E. Martinez, S. X. Wang, S. Quint, M. F. Brandl, V. Nebendahl, C. F. Roos, M. Chwalla, M. Hennrich, and R. Blatt, New J. Phys., **15**, 123012 (2013).

[10] A. M. Meier, B. Eastin, and E. Knill, Quant. Info. and Comp., **13**, 0195 (2013), arXiv:1204.4221.

[11] S. Bravyi and J. Haah, Phys. Rev. A, **86**, 052329 (2012).

[12] A. Landahl and C. Cesare, "Complex instruction set computing architecture for performing accurate quantum z rotations with less magic," (2013), arXiv:1302.3240.

[13] V. Kliuchnikov, D. Maslov, and M. Mosca, Phys. Rev. Lett., **110**, 190502 (2013), arXiv:1212.6964.

k\#rounds	0	1	2	3
4	(2, 1)	(3.46, 49.3)	(6.83, 233)	(13.6, 1.11e3)
5	(2, 1)	(3.50, 73.2)	(6.78, 351)	(13.5, 1.70e3)
6	(2, 1)	(3.53, 97.4)	(6.76, 482)	(13.4, 2.36e3)
7	(2, 1)	(3.55, 122)	(6.75, 626)	(13.3, 3.10e3)
8	(2, 1)	(3.57, 147)	(6.74, 782)	(13.3, 3.93e3)
9	(2.21, 0)	(3.80, 168)	(6.87, 946)	(13.4, 4.84e3)
10	(2.51, 0)	(3.91, 191)	(6.93, 1.12e3)	(13.4, 5.85e3)
15	(4.02, 0)	(7.17, 1.67e3)	(13.8, 1.18e4)	
20	(5.52, 0)	(7.26, 2.02e3)	(14.0, 1.84e4)	
25	(7.03, 0)	(14.3, 2.30e4)		
30	(8.53, 0)	(14.6,2.48e4)		
35	(10.0, 0)	(14.7,2.66e4)		
40	(11.5, 0)	(14.7, 2.84e4)		
45	(13.0, 0)	(14.9, 1.53e4)		

TABLE I. A sample of the data used in Figure 3. The table lists the pairs $\left(\log_{10}(1/\delta), \text{Cost}\right)$ as a function of k and the number of rounds of distillation.

(k, #rounds)	$\|Y_3\rangle$ for Λ(Swap)	$\{\|Y_{k-1}, \ldots, \|Y_3\rangle\rangle\}$
(4,0)	1	{1}
(4,1)	2	{4}
(4,2)	4	{7}
(5,0)	1	{1,1}
(5,1)	2	{2,4}
(5,2)	4	{3,7}
(6,0)	1	{1,1,1}
(6,1)	2	{2,2,4}
(6,2)	4	{3,3,7}
(9,0)	1	{1,1,1,1,1,1}
(9,1)	2	{2,2,2,2,2,4}
(9,2)	4	{3,3,3,3,3,7}
(20,1)	4	{2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3,8}

TABLE II. A sample of the states used to perform distillation recursively.

[14] A. Bocharov, M. Roetteler, and K. M. Svore, "Efficient synthesis of universal repeat-until-success circuits," (2014), arXiv:1404.5320.

[15] N. Ross and P. Selinger, "Optimal ancilla-free Clifford+T approximation of z-rotations," (2014), arXiv:1403.2975.

[16] H. Bombin, New J. Phys., **13**, 043005 (2011), arXiv:1006.5260.

[17] G. Nebe, E. M. Rains, and N. J. A. Sloane, Designs, Codes and Cryptography, **24**, 99 (2001), ISSN 0925-1022, 1573-7586.

[18] S. Bravyi and A. Kitaev, Phys. Rev. A, **71**, 022316 (2005).

[19] G. Duclos-Cianci and K. M. Svore, Phys. Rev. A., **88**, 042325 (2013), arXiv:1210.1980.

[20] C. Jones, "Distillation protocols for Fourier states in quantum computing," arXiv:1303.3066 (2013).

[21] R. Solovay, MSRI Conference on the Mathematics of Quantum Computation, Berkeley, CA (2000).

[22] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, "Can quantum chemistry be performed on a small quantum computer?" (2013), arXiv:1312.1695.

[23] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, "Improving quantum algorithms for quantum chemistry," (2014), arXiv:1403.1539.

[24] D. Poulin, M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, "The trotter step size required for accurate quantum simulation of quantum chemistry," (2014), arXiv:1406.4920.

[25] D. Gottesman and I. Chuang, Nature, **402**, 390 (1999).

[26] A. Paetznick and K. M. Svore, "Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries," (2013), arXiv:1311.1074.

[27] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2000).

[28] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo, "An algorithm for the t-count," (2013), arXiv:1308.4134.

[29] N. Cody Jones , J.D. Whitfield , P.L. McMahon , M.-H. Young , R. Van Meter , A. Aspuru-Guzik and Y. Yamamoto , "Simulating chemistry efficiently on fault-tolerant quantum computers," New Journal of Physics, **14**, 115023 (2012), arXiv:1204.0567.

[30] N. Wiebe and M. Roetteler, "Quantum arithmetic and numerical analysis using Repeat-Until-Success circuits," arXiv:1204.0567.

Chapitre 4

Article : Fault-tolerant conversion between the Steane and Reed-Muller codes.

Jonas T. Anderson, Guillaume Duclos-Cianci, David Poulin, *Fault-Tolerant Conversion between the Steane and Reed-Muller Quantum Codes*, Phys. Rev. Lett. 113, 080501 (2014).

4.1 Contexte

Ces travaux ont vu le jour durant l'étude de l'article « Using concatenated quantum codes for universal fault-tolerant quantum gates » [1]. En effet, c'est l'interprétation que j'en ai faite qui a lancé la réflexion. J'ai pu reformuler leur approche comme un décodage partiel suivi de l'application d'une porte non-encodée, puis d'un ré-encodage. Nous avons pu généraliser cette idée pour convertir le code de Steane en différents codes de Reed-Muller de manière tolérante aux fautes. L'intérêt vient du fait que le code de Steane permet d'appliquer le groupe de Clifford alors que les codes de Reed-Muller permettent de réaliser des rotations d'angles inférieurs à $\pi/2$. Toutes ces portes forment un ensemble universel. L'idée est donc de passer efficacement et de manière tolérante aux fautes d'un code à l'autre en fonction de la prochaine porte à effectuer. Cette approche ne nécessite donc pas d'injection d'états magiques. J'ai participé à l'élaboration des transformations permettant ces conversions de codes.

4.2 Résumé

Dans la section *Codes*, nous révisons les codes linéaires classiques et leurs analogues quantiques, les codes stabilisateurs. La section *The Reed-Muller code* définit de manière récursive la famille des codes de Reed-Muller. Plusieurs de leurs propriétés importantes y sont aussi énumérées. À l'aide de ceux-ci, nous montrons comment construire les codes de Reed-Muller quantiques. Les générateurs du code de Reed-Muller quantique à 15 qubits et du code à sept qubits de Steane sont donnés aux Tab. 4.1 et 4.2, respectivement, de la section 4.3. Dans la section *Transversal Gates*, nous présentons l'ensemble des portes transverses de ces codes. Finalement, la section *Conversion* présente le protocole de conversion à proprement parler qui permet de passer du code à sept qubits de Steane au code de Reed-Muller à 15 qubits.

4.3 Commentaire sur l'article « Using concatenated quantum codes for universal fault-tolerant quantum gates »

Dans la mesure où l'article « Using concatenated quantum codes for universal fault-tolerant quantum gates » [1] a inspiré les travaux présentés dans le nôtre, il est pertinent de mettre en évidence l'interprétation que nous en avons faite. Dans cet article, les auteurs proposent d'utiliser deux codes, le code de Steane et le code de Reed-Muller à 15 qubits et de les concaténer dans le but de réaliser de manière tolérante aux fautes un ensemble universel de portes.

Les Tab. 4.1 et 4.2 présentent respectivement les opérateurs définissants le code de Reed-Muller quantique à 15 qubits $\big(\mathrm{QRM}(15)\big)$ et le code à sept qubits de Steane, qui correspond en fait au QRM(7). Les opérateurs chapeautés d'un trait sont des opérateurs logiques. Dans le cas qui nous concerne, QRM(15) est concaténé au QRM(7), c'est pourquoi les opérateurs à un qubit du Tab. 4.2 apparaissent tous avec un trait et c'est pourquoi les opérateurs logiques sont chapeautés de deux traits, c.-à-d. qu'ils sont doublement encodés. De plus, nous avons placé dans chacun des tableaux des boîtes mettant en évidence la nature récursive des codes QRM définis à partir des équations 2, 3 et 4 de notre article.

Énumérons les portes transverses de ces deux codes. QRM(7) a les portes transverses $\overline{H} = H^{\otimes 7}$, $\overline{\Lambda(X)} = \Lambda(X)^{\otimes 7}$ et $\overline{S} = S^{\otimes 7}$. Ces portes génèrent le groupe de Clifford. QRM(15) a plutôt les portes $\overline{\Lambda(X)} = \Lambda(X)^{\otimes 15}$ et $\overline{T} = T^{\otimes 15}$ $\left(T = \sqrt{S}\right)$ transverses. Or, T combiné au groupe de Clifford est universel. À partir de là, voyons comment les différentes portes sont

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_1	X	X	.	X	X	.	.	X	X	.	X	X	.	.	.
S_2	.	X	X	.	X	X	.	.	X	X	.	X	X	.	.
S_3	.	.	.	X	X	X	X	.	.	.	X	X	X	X	.
S_4	.	.	.	.	.	.	.	X	X	X	X	X	X	X	X
S_5	Z	Z	.	Z	Z	.	.	.	.	.	.	.	.	.	.
S_6	.	Z	Z	.	Z	Z	.	.	.	.	.	.	.	.	.
S_7	.	.	.	Z	Z	Z	Z	.	.	.	.	.	.	.	.
S_8	.	.	.	.	.	.	.	Z	Z	.	Z	Z	.	.	.
S_9	.	.	.	.	.	.	.	.	Z	Z	.	Z	Z	.	.
S_{10}	.	.	.	.	.	.	.	.	.	.	Z	Z	Z	Z	.
S_{11}	.	.	.	.	.	.	.	Z	Z	Z	.	.	.	.	Z
S_{12}	Z	Z	.	.	.	.	.	Z	Z	.	.	.	.	.	.
S_{13}	.	Z	Z	.	.	.	.	.	Z	Z	.	.	.	.	.
S_{14}	Z	.	.	Z	.	.	.	Z	.	.	Z	.	.	.	.
$\overline{X}$	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
$\overline{Z}$	Z	Z	Z	.	.	.	.	.	.	.	.	.	.	.	.

Tableau 4.1 Générateurs du stabilisateur et opérateurs logiques du code de Reed-Muller quantique à 15 qubits.

	1	2	3	4	5	6	7
S_1	$\overline{X}$	$\overline{X}$	.	$\overline{X}$	$\overline{X}$	.	.
S_2	.	$\overline{X}$	$\overline{X}$	.	$\overline{X}$	$\overline{X}$	.
S_3	.	.	.	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$
S_4	$\overline{Z}$	$\overline{Z}$	.	$\overline{Z}$	$\overline{Z}$	.	.
S_5	.	$\overline{Z}$	$\overline{Z}$	.	$\overline{Z}$	$\overline{Z}$	.
S_6	.	.	.	$\overline{Z}$	$\overline{Z}$	$\overline{Z}$	$\overline{Z}$
$\overline{\overline{X}}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$
$\overline{\overline{Z}}$	$\overline{Z}$	$\overline{Z}$	$\overline{Z}$	.	.	.	.

Tableau 4.2 Générateurs du stabilisateur et opérateurs logiques du code de Steane à sept qubits, concaténé au code du Tab. 4.1, résultant en un code à 105 qubits.

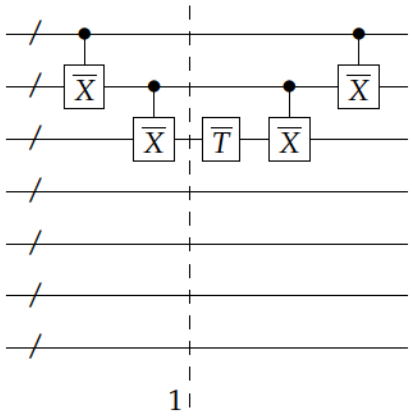


FIGURE 4.1 Reproduction de la Figure 2 de l'article [1]. Ce circuit permet d'appliquer la porte $\overline{\overline{T}}$ de manière tolérante aux fautes.

appliquées sur le code concaténé. Premièrement nous avons $\overline{\overline{\Lambda(X)}} = \overline{\Lambda(X)}^{\otimes 7} = \Lambda(X)^{\otimes 105}$, c.-à-d. que cette porte est transverse pour le code concaténé aussi. Deuxièmement, $\overline{\overline{H}} = \overline{H}^{\otimes 7}$, mais $\overline{H}$ n'est pas transverse pour QRM(15). Nous décidons de l'appliquer sur chaque copie de QRM(15) d'une manière qui n'est pas tolérante aux fautes. [1] Or, ceci n'est pas un problème, car une seule erreur physique dans l'application de $\overline{\overline{H}} = \overline{H}^{\otimes 7}$ ne peut se traduire que par une seule porte $\overline{H}$ erronée. QRM(7) peut donc la corriger. Troisièmement, $\overline{\overline{T}}$ est appliqué à l'aide du circuit présenté à la Fig. 4.1. Au point 1 du circuit, le code est modifié par les deux opérations $\overline{\Lambda(X)}$. Les opérateurs décrivant le code résultant sont énumérés au Tab. 4.3. Remarquons que l'opérateur logique $\overline{\overline{Z}}$ a été partiellement décodé pour coïncider avec l'opérateur $\overline{Z}$ du troisième QRM(15). Il s'ensuit que

$$\overline{\overline{T}} = \exp(-i\overline{\overline{Z}}\pi/4) = \exp(-i\overline{Z}_3\pi/4) = \overline{T}_3. \tag{4.1}$$

Or, $\overline{T}$ est une porte transverse de QRM(15). Elle peut donc être appliquée de manière tolérante aux fautes. Le code est finalement « ré-encodé » dans sa forme initiale. De plus, si des erreurs s'introduisent lors de l'application des $\Lambda(X)$, les codes QRM(15) affectés peuvent les corriger. Ces trois portes, $\{H, T, \Lambda(X)\}$, forment un ensemble universel.

Le point clé est de reconnaître que le circuit de la Fig. 4.1 ne doit pas être interprété comme une seule porte effective. Il doit en fait être compris en trois étapes. Tout d'abord, nous passons du code concaténé au code du Tab. 4.3. Ensuite, nous appliquons la porte $\overline{\overline{T}}$ de ce nouveau code. Finalement, nous retournons au code concaténé. Une fois que nous avons cette perspective, nous pouvons nous demander si cette procédure est efficace. En effet, elle

1. Une façon naive d'y parvenir serait de tout simplement de décoder le code à 15 qubits, d'appliquer H au qubit logique, puis de ré-encoder.

	1	2	3	4	5	6	7
S_1	$\overline{X}$	.	.	$\overline{X}$	$\overline{X}$	.	.
S_2	.	$\overline{X}$	.	.	$\overline{X}$	$\overline{X}$	.
S_3	.	.	.	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$
S_4	.	$\overline{Z}$	.	$\overline{Z}$	$\overline{Z}$	.	.
S_5	$\overline{Z}$	.	$\overline{Z}$	.	$\overline{Z}$	$\overline{Z}$	.
S_6	.	.	.	$\overline{Z}$	$\overline{Z}$	$\overline{Z}$	$\overline{Z}$
$\overline{\overline{X}}$	$\overline{X}$	.	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$	$\overline{X}$
$\overline{\overline{Z}}$	.	.	$\overline{Z}$	.	.	.	.

Tableau 4.3 Générateurs du stabilsateur et opérateurs logiques du code intermédiaire obtenu à la ligne pointillée 1 de la Fig. 4.1. Il s'agit toujours d'un code à 105 qubits.

utilise 105 qubits, ce qui motive la recherche d'une méthode moins gourmande en termes de qubits physiques. Or, les travaux présentés dans notre article répondent à cette question par l'affirmative. Nous avons montré qu'il est possible de passer directement du QRM(7) au QRM(15) de manière tolérante aux fautes, évitant ainsi complètement la concaténation. Le nombre de qubits requis est donc 15 plutôt que 105. De plus, comme nous l'avons vu ci-haut, l'ensemble des portes tolérantes aux fautes applicables sur ces deux codes est universel.

4.4 Article

Fault-tolerant conversion between the Steane and Reed-Muller quantum codes

Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin
Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, Canada
(Dated: June 27, 2014)

Steane's 7-qubit quantum error-correcting code admits a set of fault-tolerant gates that generate the Clifford group, which in itself is not universal for quantum computation. The 15-qubit Reed-Muller code also does not admit a universal fault-tolerant gate set but possesses fault-tolerant T and control-control-Z gates. Combined with the Clifford group, either of these two gates generate a universal set. Here, we combine these two features by demonstrating how to fault-tolerantly convert between these two codes, providing a new method to realize universal fault-tolerant quantum computation. One interpretation of our result is that both codes correspond to the same subsystem code in different gauges. Our scheme extends to the entire family of quantum Reed-Muller codes.

A prominent technique of fault-tolerant quantum computation is the use of transversal gates [1]. In an architecture where each logical qubit is encoded in a code block which can protect against up to t errors, a gate is said to be transversal if it does not couple qubits inside a given block. As a consequence of transversality, the number of errors or faults in a block cannot increase under the application of a gate: the number of errors after the gate is at most the number of initial errors on the data plus the number of faults in the execution of the gate. Single-qubit errors can propagate to single-qubit errors in other blocks, but these will be corrected independently on each block. In this way, an error-rate ϵ becomes $c\epsilon^{t+1}$ after error-correction, where c is at most the number of different ways of getting $t+1$ faults in a single block. Recursing this procedure leads to the celebrated accuracy threshold theorem [1–5].

Unfortunately, it is not possible to construct a quantum code which admits a universal set of transversal gates [6], so additional techniques are required. In many circumstances it is possible to fault-tolerantly implement the Clifford group, a finite sub-group of the unitary group which is not universal. In particular, all codes of the CSS family have transversal controlled-not operations [7], and code deformation can be used to implement the entire Clifford group in topological codes [8]. Magic-state distillation and injection [9] is the most common technique to complete the universal gate set.

Recently, other techniques have been proposed to circumvent this no-go on transversal gates. Jochym-O'Connor and Laflamme [10] used a "relaxed" notion of transversality which only demands that gates do not transform a single error or fault into an uncorrectable error, without prohibiting that it couples qubits from the same block. The same idea is responsible for the success of code deformation [8, 11], which changes the error-correcting code in such a way that a full cycle returning to the original code implements a gate. Because each step in the deformation acts on a number of qubits which is less than the minimum distance of the codes, the transformation is fault-tolerant despite being non-transversal [12]. Therefore, schemes for topological quantum computation [13] are a form of code deformation. Paetznick and Reichardt [14] (see also a related idea of Knill, Laflamme, and Zurek [15]) proposed a scheme where transversal gates take the system outside the code space, but a subsequent round of error correction restores it. As we discuss below, this is conceptually equivalent to Bombín's scheme [16] where transversal gates are applied to a subsystem code [17, 18], altering the gauge degrees of freedom while applying a logical gate to the encoded data. The gauge can be returned to a standard state before a new gate is applied.

Here, we propose a scheme that converts between two codes which, jointly, possess a universal set of transversal gates. Clifford group transformations are realized in Steane's 7-qubit code [19], while the $T = Z^{1/4}$ gate and/or the control-control-Z gate are realized using the 15-qubit Reed-Muller code [15]; either of these last two gates is sufficient to complete the universal gate set, but an over-complete set can reduce the compilation overhead [20]. While it is always possible to convert between codes by preparing a special ancillary entangled state to teleport the data, our main contribution is a fault-tolerant scheme which directly converts the information in place. Much like in the approaches outlined above, the code is modified during the computation. One important difference here is that the codes involved have different numbers of qubits, an aspect that should be taken into account when optimizing resources to realize a given quantum circuit. Similarly to the proposals of [14] and [16], our scheme can be seen as a subsystem encoding [17, 18] with different gauge fixing. In fact, our approach should be seen as complementary to [14, 16], which enables a much richer set of transversal gates and extends to the entire quantum Reed-Muller code family.

The rest of this paper is organized as follows. After a brief review of classical and quantum codes, we present the family of quantum Reed-Muller codes and highlight some of their key properties. Then, we review transversal gate constructions for these codes. We then explain the conversion scheme, which essentially relies on a recursive definition of the Reed-Muller codes. Lastly, we present an alternative derivation in terms of subsystem codes, and conclude by discussing possible applications of our scheme.

Codes— An n-bit classical linear code encoding k bits

is defined as the null-space of a $(n-k)\times n$ parity-check matrix H (in $\mathbb{Z}_2$ arithmetic), i.e. $\mathcal{C}=\{x\in\mathbb{Z}_2^n : Hx=0\}$. Its minimum distance d is the minimum number of bit-flips required to map one code-word to another. Given an erroneous string $x'=x+e$ obtained from a code word x and error e, the error syndrome is given by $s=Hx'=He$ and can unambiguously identify any error acting on less than or equal to $(d-1)/2$ bits. The code can also be defined as the row-space of a $k\times n$ generator matrix G, i.e. $\mathcal{C}=\mathrm{row}(G)$, which is the dual of H, i.e., $HG^T=0$.

A stabilizer code encoding k qubits into n qubits is specified by a set $\mathcal{A}$ of $n-k$ independent stabilizer generators, which are commuting and hermitian elements of the n-qubit Pauli group (obtained from n-fold tensor product of the 2×2 identity I and the Pauli matrices X, Y, and Z). The code space $\mathcal{C}$ is a subspace of the n-qubit Hilbert space stabilized by $\mathcal{A}$:

$$\mathcal{C}=\{|\psi\rangle\in(\mathbb{C}^2)^{\otimes n} : A|\psi\rangle=|\psi\rangle \quad \forall A\in\mathcal{A}\}. \tag{1}$$

Equivalently, it can be defined as the image of the code projector $P_{\mathcal{A}}=\prod_{A\in\mathcal{A}}\frac{I+A}{2}=\frac{1}{2^{|\mathcal{A}|}}\sum_{S\in\mathcal{S}}S$ where $\mathcal{S}$ is the stabilizer group generated by $\mathcal{A}$. When a code state $|\psi\rangle\in\mathcal{C}$ undergoes a Pauli error E, error correction is realized by measuring the stabilizer generators. The ±1 measurement outcome of measuring $A_j\in\mathcal{A}$ indicates whether A_j commutes or anti-commutes with E: $A_j(E|\psi\rangle)=\pm EA_j|\psi\rangle=\pm(E|\psi\rangle)$. Logical operators transform the state but preserve the code space, i.e. they are elements of $N(\mathcal{S})-\mathcal{S}$, where N denotes the normalizer of a group. A code has distance d if it takes an error of weight d or more to map a codeword to a distinct codeword. These parameters of a code are collectively denoted (n,k,d) in the classical setting and $[[n,k,d]]$ in the quantum setting.

The Reed-Muller code— The Reed-Muller codes of order 1 can be defined recursively [21]: the code $\mathsf{RM}(1,1)$ has generator matrix

$$G_1=\begin{pmatrix}1&1\\0&1\end{pmatrix} \tag{2}$$

and the code $\mathsf{RM}(1,m+1)$ has generator matrix

$$G_{m+1}=\begin{pmatrix}G_m&G_m\\\mathbf{0}&\mathbf{1}\end{pmatrix}. \tag{3}$$

(and bold symbols $\mathbf{0}$ and $\mathbf{1}$ designate strings of 0s and 1s of lengths fixed by the context). The dual of $\mathsf{RM}(1,m)$ is $\mathsf{RM}(m-2,m)$ and has generator matrix H_m. Quantum codes are derived from shortened Reed-Muller codes $\overline{\mathsf{RM}}(1,m)$, where the first row and column are deleted from G_m. We can similarly define shortened dual codes $\overline{\mathsf{RM}}(m-2,m)$ with generator matrix $\overline{H}_m$. Hence, the generator matrices of $\overline{\mathsf{RM}}(1,m)$ obey the recursive definition

$$\overline{G}_{m+1}=\begin{pmatrix}\overline{G}_m&\overline{G}_m&\mathbf{0}^T\\\mathbf{0}&\mathbf{1}&1\end{pmatrix} \tag{4}$$

(we have permuted the columns for later convenience). Note that $\overline{\mathsf{RM}}(m-2,m)$ is not the dual of $\overline{\mathsf{RM}}(1,m)$. Using this definition, the following Facts can easily be verified (see Appendix A) by induction for $m\geq 2$:

1. For $x\in\mathsf{RM}(1,m)$ or $\overline{\mathsf{RM}}(1,m)$, $|x|=0 \mod 2^{m-1}$.
2. For $m\geq 3$, $\overline{\mathsf{RM}}(1,m)$ is contained in its dual, i.e., $\overline{G}_m\overline{G}_m^T=0$.
3. The minimum distance of the dual code to $\overline{\mathsf{RM}}(1,m)$ is 3.
4. $\overline{\mathsf{RM}}(1,m)$ is contained in the dual of $\overline{\mathsf{RM}}(m-2,m)$, i.e. $\overline{H}_m\overline{G}_m^T=0$.
5. $\overline{\mathsf{RM}}(1,m)$ is contained in $\overline{\mathsf{RM}}(m-2,m)$, i.e. $\mathrm{row}(\overline{G}_m)\subset\mathrm{row}(\overline{H}_m)$.
6. For $x_1,x_2,\ldots x_p\in\overline{\mathsf{RM}}(1,m)$, $x_1\cdot x_2\cdot\ldots x_p=0 \mod 2^{m-p}$.

The quantum Reed-Muller codes [22] $\mathsf{QRM}(m)$ derived from $\overline{\mathsf{RM}}(1,m)$ codes are CSS codes, meaning that their stabilizer generators break into two sets $\mathcal{A}_m^x$ and $\mathcal{A}_m^z$ [23, 24]. Elements of $\mathcal{A}_m^x$ are obtained from rows of $\overline{G}_m$, by substituting 1s by Xs and 0s for Is. Elements of $\mathcal{A}_m^z$ are obtained in a similar way, but from the generator matrix of the shortened dual code $\overline{\mathsf{RM}}(m-2,m)$.

In a CSS code, $\mathcal{A}^x$ detects z-type errors and $\mathcal{A}^z$ detects x-type errors. It follows from Fact 3 that $\mathsf{QRM}(m)$ have minimum distance $d=3$, so the parameters of the code are $[[n=2^m-1,k=1,d=3]]$. The logical operators are given by the rows that were removed in the shortening procedure, they are $\overline{X}_m=X^{\otimes n}$ and $\overline{Z}_m=Z^{\otimes n}$. Finally, note that the commutation of the stabilizer generators follows from the orthogonality of $\overline{\mathsf{RM}}(1,m)$ and $\overline{\mathsf{RM}}(m-2,m)$ (Fact 4).

Transversal gates — The logical 0 state of a code should be a simultaneous $+1$ eigenstate of $\overline{Z}$ and all elements of $\mathcal{A}$. The state $|\mathbf{0}\rangle$ is already a $+1$ eigenstate of $\overline{Z}$ and of all $\mathcal{A}_m^z$, so we obtain the logical 0 by projecting it onto the $+1$ eigenspace of elements of $\mathcal{A}_m^x$:

$$|\overline{0}\rangle_S=\prod_{A\in\mathcal{A}_m^x}\frac{I+A}{2}|\mathbf{0}\rangle \tag{5}$$

$$=\frac{1}{|\mathcal{S}_x|}\sum_{S\in\mathcal{S}_m^x}S|\mathbf{0}\rangle \tag{6}$$

$$=\frac{1}{|\mathcal{S}_x|}\sum_{x\in\mathrm{row}(\overline{G}_m)}|x\rangle. \tag{7}$$

The logical 1 is obtained by applying $\overline{X}_m$ to this state, so it is $|\overline{1}\rangle=\frac{1}{|\mathcal{S}_x|}\sum_{x\in\mathrm{row}(\overline{G}_m)}|x\oplus\mathbf{1}\rangle$. It follows from Fact 1 that $|\overline{0}\rangle$ is the superposition of strings of weight $0 \mod 2^{m-1}$ and $|\overline{1}\rangle$ is the superposition of strings of weight $-1 \mod 2^{m-1}$.

Consider now the single-qubit gate $Z(\omega_\ell)=\mathrm{diag}(1,\omega_\ell)$ where ω_ℓ is the ℓth root of unity. Observe that for any

n-bit string x, $Z(\omega_\ell)^{\otimes n}|x\rangle = \omega_\ell^{|x|}|x\rangle = \omega_\ell^{(|x| \mod \ell)}|x\rangle$. From the above consideration on the weights of the basis states appearing in the logical states $|\overline{0}\rangle$ and $|\overline{1}\rangle$, it follows that for $\ell = 2^{m-1}$, the transversal gate $Z(\omega_\ell)^{\otimes n}$ acts as the logical $Z(\omega_\ell)^\dagger$ on $\mathsf{QRM}(m)$ [25–27].

The codes $\mathsf{QRM}(m)$ also have a transversal k-fold controlled-Z gate for $k \leq m-2$. Note that the transversal k-fold controlled gate acts on a basis state $|x_1\rangle|x_2\rangle\ldots|x_{k+1}\rangle$ by introduction of a phase factor $(-1)^{x_1 \cdot x_2 \cdot \ldots x_{k+1}}$. A logical state $|\bar{y}\rangle$ is the superposition of states of the form $|x + y\mathbf{1}\rangle$ where $x \in \overline{\mathsf{RM}}(1,m)$. When acted on by a transversal k-fold controlled-Z gate, a logical state $|\bar{y}_1\rangle|\bar{y}_2\rangle\ldots|\bar{y}_{k+1}\rangle$ will pick up a phase factor $(x_1 + y_1\mathbf{1}) \cdot (x_2 + y_2\mathbf{1}) \cdot \ldots (x_{k+1} + y_{k+1}\mathbf{1})$ where $x_j \in \overline{\mathsf{RM}}(1,m)$ for all j. Expanding this product, all terms containing xs produce a trivial phase due to Fact 6, so only the term $y_1 y_2 \ldots y_{k+1}$ contributes to the phase, which produces the desired transformation.

The 7-qubit Steane code is derived from the classical code $\overline{\mathsf{RM}}(1,3)$, a.k.a. the classical (7,4,3) Hamming code. This is a special case as it is self-dual, which implies that $\mathcal{A}_3^x$ and $\mathcal{A}_3^z$ are equal up to exchanging Xs for Zs. As a consequence it has transversal Clifford gates. The Hadamard gate H exchanges X and Z. It is thus clear that the transversal gate $H^{\otimes 7}$ preserves the code space (as it only swaps $\mathcal{A}_3^x$ with $\mathcal{A}_3^z$) and acts as the logical Hadamard by exchanging $\overline{X}$ with $\overline{Z}$. The CNOT acting on two qubits maps the operators (IX, XI, IZ, ZI) to (IX, XX, ZZ, ZI). The transversal gate $\mathrm{CNOT}^{\otimes 7}$ therefore acts on the logical operators as a logical CNOT, and maps the set of generators $\{I\mathcal{A}_3^x, \mathcal{A}_3^x I, I\mathcal{A}_3^z, \mathcal{A}_3^z I\}$ of $\mathcal{S}_3 \otimes \mathcal{S}_3$ to $\{I\mathcal{A}_3^x, \mathcal{A}_3^x\mathcal{A}_3^x, \mathcal{A}_3^z\mathcal{A}_3^z, \mathcal{A}_3^z I\}$, which is simply a different set of generators for $\mathcal{S}_3 \otimes \mathcal{S}_3$, so the code is preserved. Finally, the phase gate P corresponds to $Z(\omega_4)$ defined above and is transversal as we have seen.

Conversion— The key feature of quantum Reed-Muller codes which enables our construction, and which follows from Fact 5 $\overline{\mathsf{RM}}(1,m) \subset \overline{\mathsf{RM}}(m-2,m)$, is that $\mathcal{A}_m^z$ contains the same operators as $\mathcal{A}_m^x$ with Xs replaced by Zs, plus some additional operators. In other words, if we consider the checks $\mathcal{A}_m'^z$ obtained by replacing X by Z in $\mathcal{A}_m^x$, then $\mathcal{A}_m^z = \mathcal{A}_m'^z \cup \tilde{\mathcal{A}}_m^z$ for some set of z-stabilizer generators $\tilde{\mathcal{A}}_m^z$. Since elements of $\mathcal{A}_m^x$ can unambiguously discriminate all single-qubit z-errors, it follows that $\mathcal{A}_m'^z$ can unambiguously discriminate all single-qubit x-errors, i.e. operators from $\tilde{\mathcal{A}}_m^z$ are superfluous. Starting from the "relevant" stabilizers $\mathcal{A}_m^x$ and $\mathcal{A}_m'^z$, there are many ways to complete the list of stabilizers in order to obtain a valid error-correcting code. Our scheme will make use of this freedom to convert between different codes.

It follows from Eq. (4) that the relevant stabilizers $\mathcal{A}_m^x$ and $\mathcal{A}_m'^z$ can be defined recursively. Given two ordered sets $\mathcal{A} = \{A_1, A_2, \ldots\}$ and $\mathcal{B} = \{B_1, B_2, \ldots\}$, we introduce the notation $\mathcal{A} \times \mathcal{B} = \{A_1 \otimes B_1, A_2 \otimes B_2, \ldots\}$, and write

$$\mathcal{A}_{m+1}^x = \left\{ \begin{array}{ccccc} \mathcal{A}_m^x & \times & \mathcal{A}_m^x & \otimes & I, \\ I^{\otimes n} & \otimes & \overline{X}_m & \otimes & X \end{array} \right\}, \quad \text{and} \tag{8}$$

$$\mathcal{A}_{m+1}'^z = \left\{ \begin{array}{ccccc} \mathcal{A}_m'^z & \times & \mathcal{A}_m'^z & \otimes & I, \\ I^{\otimes n} & \otimes & \overline{Z}_m & \otimes & Z \end{array} \right\}. \tag{9}$$

Let us first explain how to convert from $\mathsf{QRM}(m)$ to $\mathsf{QRM}(m+1)$. We begin with some information encoded in an (2^m-1)-qubit state of $\mathsf{QRM}(m)$, $|\overline{\psi}\rangle_m$. We prepare a 2^m-qubit quantum state $|\Phi\rangle = \frac{1}{\sqrt{2}}(|\overline{0}\rangle_m|0\rangle + |\overline{1}\rangle_m|1\rangle)$ consisting of a maximally entangled state between a bare qubit and a qubit encoded in $\mathsf{RM}(m)$. Viewing the joint state $|\overline{\psi}\rangle_m \otimes |\Phi\rangle$ as an encoded state of a $(2^{m+1}-1)$-qubit code, we can write the generators for this "extended quantum Reed-Muller code" as

$$\begin{array}{ccccc} \mathcal{A}_m^z & \otimes & I^{\otimes n} & \otimes & I \\ \mathcal{A}_m^x & \otimes & I^{\otimes n} & \otimes & I \\ I^{\otimes n} & \otimes & \mathcal{A}_m^z & \otimes & I \\ I^{\otimes n} & \otimes & \mathcal{A}_m^x & \otimes & I \\ I^{\otimes n} & \otimes & \overline{Z}_m & \otimes & Z \\ I^{\otimes n} & \otimes & \overline{X}_m & \otimes & X \end{array} \tag{10}$$

We can change the generating set without changing the code and instead use

$$\begin{array}{ccccc} \mathcal{A}_m'^z & \times & \mathcal{A}_m'^z & \otimes & I \\ \mathcal{A}_m^x & \times & \mathcal{A}_m^x & \otimes & I \\ I^{\otimes n} & \otimes & \overline{Z}_m & \otimes & Z \\ I^{\otimes n} & \otimes & \overline{X}_m & \otimes & X \\ \tilde{\mathcal{A}}_m^z & \times & \tilde{\mathcal{A}}_m^z & \otimes & I \\ \mathcal{A}_m^z & \otimes & I^{\otimes n} & \otimes & I \\ \mathcal{A}_m^x & \otimes & I^{\otimes n} & \otimes & I \end{array} \tag{11}$$

We immediately recognize the first $2m+2$ generators of this list [first four rows of Eq. (11)] as generating the relevant stabilizers of $\mathsf{QRM}(m+1)$, i.e. $\mathcal{A}_{m+1}^x$ and $\mathcal{A}_{m+1}'^z$. Indeed, compare to Eqs. (8,9). Thus, only operators from the last three lines of Eq. (11) differ, and must be substituted by $\tilde{\mathcal{A}}_{m+1}^z$ to convert into $\mathsf{QRM}(m+1)$. In fact, only the m stabilizers of the last line are a problem, since $\tilde{\mathcal{A}}_m^z \times \tilde{\mathcal{A}}_m^z \otimes I$ and $\mathcal{A}_m^z \otimes I^{\otimes n} \otimes I \subset \tilde{\mathcal{A}}_{m+1}^z$.

But as explained in the previous paragraphs, these m stabilizers are superfluous in the sense that they are not required to diagnose single-qubit errors. Thus, if we fault-tolerantly measure all stabilizers of $\mathsf{QRM}(m+1)$ on the state $|\overline{\psi}\rangle_m \otimes |\Phi\rangle$, we can use the syndrome from the first six rows of Eq. (11) to diagnose errors, and remove any syndrome associated to the last m stabilizers by a fault-tolerant error-correction procedure (or by adapting the Pauli frame). Specifically, given a set of stabilizer generators $\mathcal{A} = \{A_1, \ldots A_{n-k}\}$ and logical operators $\mathcal{L} = \{\overline{X}_a, \ldots \overline{X}_k, \overline{Z}_1, \ldots \overline{X}_k\}$, there exists a set of "pure errors" $\mathcal{T} = \{T_1, \ldots T_{n-k}\}$ such that T_j commutes with all elements of $\mathcal{L}$, $\mathcal{T}$, and $\mathcal{A}$ except A_j with which it anti-commutes. A syndrome $A_j = -1$ revealed by one of the last m stabilizers $j = n-k-m, \ldots n-k$ is corrected by applying T_j.

To summarize, to convert from QRM(m) to QRM($m+1$), we first fault-tolerantly prepare the 2^m-qubit stabilizer state $|\Phi\rangle$, append it to the system, fault-tolerantly measure the stabilizer generators of QRM($m+1$), error-correct given the first $2^{m+1}-m-2$ syndrome bits (first six rows of Eq. (11)) and restore the last m syndrome bits using their associated pure errors.

To convert from QRM($m+1$) to QRM(m), we simply fault-tolerantly measure the stabilizers of Eq. (11), use the first $2^{m+1}-m-2$ syndrome bits (first six rows of Eq. (11)) to diagnose errors, and restore the last m syndrome bits using the associate pure errors. We can then remove the additional 2^m qubits and be left with the (2^m-1)-qubit state $|\overline{\psi}\rangle_m$ encoded in QRM(m).

Subsystem code interpretation— It is possible to recast the above conversion scheme using the subsystem code formalism [17, 18], which highlights its similarity with the Paetznick and Reichardt [14] and the Bombín [16] schemes. We can define a stabilizer code from the stabilizers that are common to QRM($m+1$) and the extended QRM(m). There are $2^{m+1}-m-2$ of these and they are given by the first six lines of Eq. (11). Thus, this code encodes $k=m+1$ logical qubits and has minimum distance $d=3$, so it can error-correct any single-qubit error.

One of these logical qubits, which we label 0, is the one encoded in the original code and has logical operators $\overline{X}^0=\overline{X}_m$ and $\overline{Z}^0=\overline{Z}_m$. The other logical operators associated to "gauge qubits", $\overline{X}^j$ with $j=1,\ldots,m$ correspond to elements of the last line of Eq. (11). Their conjugate partners $\overline{Z}^j$ are generated by elements of $\tilde{\mathcal{A}}^z_{m+1}$.

We obtain a subsystem code by choosing to encode information only in the first logical qubit of the code. The other logical qubits $j=1,2,\ldots m$ carry no information, and can be fixed to an arbitrary state. The conversion scheme described above then simply consists in fixing these m gauge qubits all in state $|\bar{0}\rangle$ or all in state $\frac{1}{\sqrt{2}}(|\bar{0}\rangle+|\bar{1}\rangle)$. The first scenario can be realized by measuring the operators $\overline{Z}^j$, and flipping the qubit using $\overline{X}^j$ if the outcome is -1. This procedure brings the state to the extended quantum Reed-Muller code, and the last 2^m qubits can be discarded to obtain a state encoded in QRM(m). The second scenario can be realized by measuring the operators $\overline{X}^j$, and flipping the qubit using $\overline{Z}^j$ if the outcome is -1. This procedure brings the state to QRM($m+1$).

Thus, we see that the different quantum Reed-Muller codes all correspond to the same subsystem code with different gauge fixing. Depending on the chosen gauge, some qubits become unentangled with the part of the code supporting the data, and can be discarded. At the bottom of this hierarchy is Steane's 7-qubit code, which realizes the entire Clifford group transversally. Above is an infinite family of quantum Reed-Muller codes which admit increasingly complex transversal gates [28].

Conclusion & Outlook— We have presented a scheme to directly and fault-tolerantly convert between a family of quantum error correcting codes. By combining the transversal gate sets of these codes, we obtain a (overcomplete) universal gate set. Our result offers a deeper understanding of a recent proposal [14] and complements it in many ways.

An important advantage of our conversion scheme is its potential reduction of overhead. We can envision an architecture where special areas in the computer are dedicated to the execution of non-Clifford gates. In those areas, the encoding uses concatenated Reed-Muller code, while the rest of the computer is encoded with concatenated Steane codes, an important overhead reduction over [14] when few non-Cliford gates are executed in parallel. Qubits are brought into these special areas to realize non-Clifford gates.

Finally, we note that the higher-order Reed-Muller codes RM(r,m) obey a similar recursive definition

$$G_{r,m+1}=\begin{pmatrix} G_{r,m} & G_{r,m} \\ 0 & G_{r-1,m}\end{pmatrix} \tag{12}$$

and are dual-containing when their rates is more than 1/2 [21], so our conversion procedure can be extended to this broader class of codes (see appendix B). The two main motivation to study these codes is that they can have a larger minimal distance and admit a richer set of transversal gates [27]. Moreover, the Reed-Muller code family can be used to distill magic states [14, 25–27]. Distillation is a procedure which uses Clifford operations to increase the fidelity of non-stabilizer states, which can be injected in the computation to realize non-Clifford transformations [9]. Higher-order Reed-Muller codes of minimal distance greater than 3 could be used to improve magic state distillation protocols.

Acknowledgements— This work was partially funded by Canada's NSERC, Québec's FRQNT through the network INTRIQ, and the Lockheed Martin Corporation. DP acknowledges the hospitality of The University of Sydney where this project was completed. We thank Michael Beverland for comments on this manuscript.

[1] P. W. Shor, in *Proceedings of the 37th Symposium on the Foundations of Computer Science* (IEEE press, Los Alamitos, California, 1996) pp. 56–65
[2] D. Aharonov and M. Ben-Or, in *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997)
[3] A. Y. Kitaev, Russ. Math. Surv., **52**, 1191 (1997)
[4] E. Knill, R. Laflamme, and W. H. Zurek, Phil. Trans. R. Soc. Lond. A., **454**, 365 (1998)
[5] J. Preskill, Proc. R. Soc. Lond. A, **454**, 385 (1998)

[6] B. Eastin and E. Knill, Phys. Rev. Lett., **102**, 110502 (2009)
[7] A. M. Steane, Nature, **399**, 124 (1999)
[8] H. Bombin and M. Martin-Delgado, J. Phys. A, **42** (2009)
[9] S. Bravyi and A. Kitaev, Phys. Rev. A, **71**, 022316 (2005)
[10] T. Jochym-O'Connor and R. Laflamme, Phys. Rev. Lett., **112**, 010505 (2014)
[11] H. Bombin, New J. Phys., **13**, 043005 (2011)
[12] H. Bombin, G. Duclos-Cianci, and D. Poulin, New J. Phys., **14**, 073048 (2012)
[13] R. Raussendorf and J. Harrington, Phys. Rev. Lett., **98**, 190504 (2007)
[14] A. Paetznick and B. W. Reichardt, Phys. Rev. Lett., **111**, 090505 (2013)
[15] E. Knill, R. Laflamme, and W. Zurek, "Accuracy threshold for quantum computation," (1996), quant-ph/9610011
[16] H. Bombín, "Optimal transversal gates under geometric constraints," (2013), arXiv:1311.0879
[17] D. Kribs, R. Laflamme, and D. Poulin, Phys. Rev. Lett., **94**, 180501 (2005)
[18] D. Poulin, Phys. Rev. Lett., **95**, 230504 (2005)
[19] A. M. Steane, Phys. Rev. Lett., **77**, 793 (1996)
[20] G. Duclos-Cianci and D. Poulin, "Reducing the quantum computing overhead with complex gate distillation," (2014), arXiv:1403.5280
[21] F. J. MacWilliams and N. J. Sloane, *The Theory of Error-Correcting Codes* (North-Holland Publishing Company, 1977)
[22] A. M. Steane, IEEE Trans. Info. Theor., **45**, 1701 (1999)
[23] A. R. Calderbank and P. W. Shor, Phys. Rev. A, **54**, 1098 (1996)
[24] A. M. Steane, Proc. R. Soc. Lond. A, **452**, 2551 (1996)
[25] E. T. Campbell, H. Anwar, and D. E. Browne, Phys. Rev. X, **2**, 041021 (2012)
[26] S. Bravyi and J. Haah, Phys. Rev. A, **86**, 052329 (2012)
[27] A. Landahl and C. Cesare, "Complex instruction set computing architecture for performing accurate quantum z rotations with less magic," (2013), arXiv:1302.3240
[28] D. Gottesman and I. Chuang, Nature, **402**, 390 (1999)

Supplementary Information: Fault-tolerant conversion between the Steane and Reed-Muller quantum codes

Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin*
Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, Canada
(Dated: June 18, 2014)

Appendix A— In this appendix we prove the 6 properties of the shortened Reed-Muller codes listed in the main text as Facts. It will be useful to make use of an alternative recursive definition of these codes [1]:

$$\mathsf{RM}(1,m+1) = \{(x,x),(x,x+\mathbf{1}) : x \in \mathsf{RM}(1,m)\}\,. \quad (1)$$

Fact 1. For $\mathsf{RM}(1,m)$, the base case $m=2$ can be verified directly. Suppose that the fact holds for m, which means that the allowed weights of elements of $\mathsf{RM}(1,m)$ are $w_m = 0$, 2^{m-1}, or 2^m. Using Eq. (1), we see that the weight of elements of $\mathsf{RM}(1,m+1)$ will be either $2w_m$ or $w_m + (2^m - w_m)$, so the condition is satisfied. When we shorten the code to get $\overline{\mathsf{RM}}(1,m)$, we remove a row from G_m which contains all 1s and then remove a column containing all 0s. Thus, we have

$$\mathsf{RM}(1,m) = \left\{(0,x),(1,x+\mathbf{1}) : x \in \overline{\mathsf{RM}}(1,m)\right\}. \quad (2)$$

Thus, the set $\{(0,x) : x \in \overline{\mathsf{RM}}(1,m)\}$ is a subset of $\mathsf{RM}(1,m)$, so the property holds for $\overline{\mathsf{RM}}(1,m)$ as well.

Fact 2. The base case $m=3$ is well known, it corresponds to the Hamming code (Steane's code). The induction yields

$$\overline{G}_{m+1}\overline{G}_{m+1}^T = \begin{pmatrix} 0 & \overline{G}_m \cdot \mathbf{1}^T \\ \mathbf{1}\cdot\overline{G}_m^T & 0 \end{pmatrix}. \quad (3)$$

Noting that $\overline{G}_m \cdot \mathbf{1}^T$ is simply the vector of weights mod 2 of the rows of $\overline{G}_m$ and that these are even by Fact 1 proves Fact 2.

Fact 3. Since we are interested in the dual code, we should think of G_m as the parity check matrix of a code. The base case $m=2$ corresponds to the parity-check matrix

$$\overline{G}_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}. \quad (4)$$

The minimum distance is obviously bounded by the length of the code $d \leq 3$. This parity-check matrix can uniquely identify any single bit error since all its columns are distinct, so it has minimum-distance 3. In its recursive definition Eq. (4, main text), $\overline{G}_{m+1}$ contains three blocks of bits: the first two of size $2^m - 1$ and the last of size 1. It is clear that the minimum distance for $m+1$ is no greater than the minimum distance for m, since an error occurring in the first block is only seen by $\overline{G}_m$. On the other hand, a single-bit error occurring in different blocks will trigger different syndrome patterns. If it is in block 1 its first m syndrome bits will be non-trivial and its last syndrome bit will be trivial. If it is in block 2 its first m syndrome bits will be non-trivial and its last syndrome bit will be non-trivial. If it is in block 3 its first m syndrome bits will be trivial and its last syndrome bit will be non-trivial. Moreover, in each case the syndrome can uniquely identify the error by induction, proving the fact.

Fact 4. To prove this fact it is important to know that for shortening, the row which is deleted from H_m is all 1s and that the subsequently deleted column is all 0s. The fact that H_m contains an all 1s row simply reflects the fact that elements of $\mathsf{RM}(1,m)$ have even weight for $m \geq 2$. The fact that the rest of the first column is all 0s can always be obtained by Gaussian elimination. By definition, $G_m H_m^T = 0$, or in other words

$$\begin{pmatrix} 1 & 1\ldots1 \\ 0 & \\ 0 & \overline{G}_m \\ 0 & \end{pmatrix}\begin{pmatrix} 1 & 0\ldots0 \\ 1 & \\ 1 & \overline{H}_m^T \\ 1 & \end{pmatrix} \quad (5)$$

$$= \begin{pmatrix} 0 & \overline{H}_m \cdot \mathbf{1}^T \\ \mathbf{1}\cdot\overline{H}_m^T & \overline{G}_m\overline{H}_m^T \end{pmatrix} = 0. \quad (6)$$

Fact 5. First, we prove that $\mathrm{row}(G_m) \subset \mathrm{row}(H_m)$. This follows from the fact that $G_m G_m^T = 0$, which we prove by induction:

$$G_{m+1}G_{m+1}^T = \begin{pmatrix} 0 & G_m \cdot \mathbf{1}^T \\ \mathbf{1}\cdot G_m^T & 0 \end{pmatrix}. \quad (7)$$

The r.h.s is 0 since rows of G_m have even weight from Fact 1. The fact follows from the observation that $\overline{G}_m$ and $\overline{H}_m$ are obtained from G_m and H_m by the same shortening procedure: first remove an all 1s row and then remove an all 0s column.

Fact 6: For this Fact it is convenient to define $\mathsf{RM}(1,m)$ as boolean polynomials with all terms of degree 1 [1]. Then, $x_1 \cdot x_2 \cdot \ldots x_p$ is a boolean polynomial with all terms of degree p, and these have weights $0 \mod 2^{m-p}$ [1].

Appendix B— In this appendix we discuss the generalization to higher rank Reed-Muller codes, defined recursively by [1]

$$\mathsf{RM}(r,m+1) = \quad (8)$$
$$\{(x,x+y) : x \in \mathsf{RM}(r,m), y \in \mathsf{RM}(r-1,m)\},$$

* David.Poulin@USherbrooke.ca

or equivalently by Eq. (12, main text).

Denote $G_{r,m}$ the generator matrix of $\mathsf{RM}(r,m)$. We choose a pair of codes $\mathsf{RM}(m-r-1,m)$ and $\mathsf{RM}(m-r,m+1)$ both of rates greater than $1/2$. Such codes contain their dual [1], so in particular the first code contains $\mathsf{RM}(r,m)$, which implies that $\mathsf{RM}(r,m)$ is self-orthogonal, and the same reasoning applies to $\mathsf{RM}(r,m+1)$. Since $\mathsf{RM}(m-r,m)$ has a rate greater than $\mathsf{RM}(m-r-1,m)$, it follows that $\mathsf{RM}(r-1,m)$ is also self-orthogonal. In short, we have just shown $G_{r,m}G_{r,m}^T = 0$, $G_{r-1,m}G_{r-1,m}^T = 0$, and $G_{r,m+1}G_{r,m+1}^T = 0$. This last equality combined to Eq. (12, main text) implies that $G_{r,m}G_{r-1,m}^T = 0$.

As a consequence of these orthogonality conditions, we can use the rows of $G_{r,m}$ to build a self-dual CSS code $\mathsf{QRM}(r,m)$. Similarly, we can build a self-dual CSS code $\mathcal{G}$ from the union of the rows of $\mathsf{RM}(r,m)$ and $\mathsf{RM}(r-1,m)$. The code $\mathcal{G}$ has minimum distance $\geq d_{r-1,m}$. There are many inequivalent ways of building subsystem codes from these, by converting some logical qubits into gauge qubits, by shortening the codes, and by adding additional stabilizers $\tilde{\mathcal{A}}_{r,m}$ or equivalently fixing the gauge in various ways. Below we briefly discuss one possible construction, which converts between two subsystem codes with stabilizers given by $\mathsf{QRM}(r,m)$ and $\mathsf{QRM}(r,m+1)$, and has minimum distance $d_{r,m} = 2^{m-r}$.

$m+1 \to m$ conversion: We begin in a subsystem code with stabilizers $\mathsf{QRM}(r,m+1)$. As of Eq. (12, main text), we can naturally partition the 2^{m+1} qubits into two blocks of 2^m qubits. We can measure the stabilizers of $\mathcal{G}$ on the second block, and correct any errors it reveals. This leaves the first block in the code $\mathsf{QRM}(r,m)$. The logical operators of $\mathsf{QRM}(r,m)$ acting on the first block are preserved by this procedure.

$m \to m+1$ conversion: We begin in the stabilizer code $\mathsf{QRM}(r,m)$. We append to the system a state ρ prepared in the code $\mathcal{G}$. The resulting state is stabilized by $\mathsf{QRM}(r,m+1)$. We can measure any additional stabilizers and use their associated pure errors to restore their $+1$ value in order to restore a given gauge. The logical operators of $\mathsf{QRM}(r,m)$ acting on the first block are preserved by this procedure provided that they do not conflict with the gauge choice.

Note that, while the conversion scheme presented here and in the main text are conceptually identical, the codes presented in the main text are not a special case of the quantum codes $\mathsf{QRM}(r,m)$ since these have not been punctured or shortened. It is an interesting open problem to study the various ways in which Reed-Muller codes can be punctured and shortened to produce quantum codes with interesting transversal gates and code parameters. We note for instance that applying the standard puncture and shortening procedure (remove the all-1 column and row) to $\mathsf{RM}(2,7)$ yields a $[[127,1,7]]$ quantum code with transversal T gate [2], which could be important for magic state distillation.

[1] F. J. MacWilliams and N. J. Sloane, *The Theory of Error-Correcting Codes* (North-Holland Publishing Company, 1977)

[2] The online encyclopedia of interer sequences http://oeis.org/a006006

Deuxième partie

Codes topologiques stabilisateurs

Chapitre 5

Article : Universal topological phase of two-dimensional stabilizer codes

Bombin H., Duclos-Cianci G., Poulin D., *Universal topological phase of two-dimensional stabilizer codes*, New Journal of Physics, 14(7), 073048 (2012).

5.1 Contexte

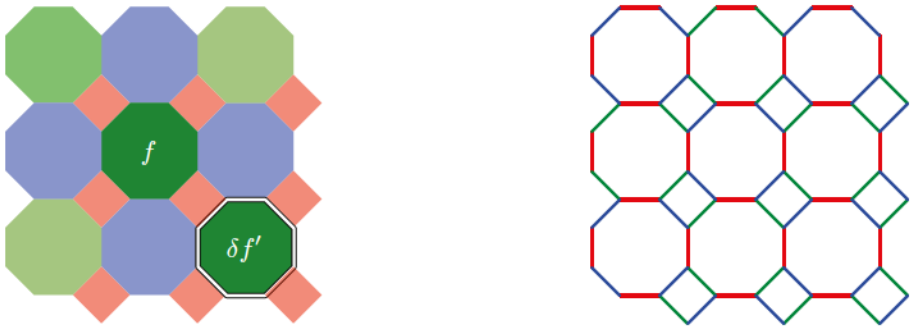
Durant l'intervalle de temps où je terminais mon mémoire de maîtrise et commençait mon doctorat, je me suis penché sur le décodage du code de couleurs. Durant cette même période, Héctor Bombin, inventeur des codes de couleurs [15, 26], visitait l'université de Sherbrooke. Nos discussions nous ont amenés à plusieurs conclusions. Tout d'abord, pour ce qui est du décodage, il peut toujours être formulé en termes d'homologie de lignes de vies d'excitations. Grâce à ce fait simple, j'arrivais à transformer les défauts du code de couleurs en défauts du code topologique de Kitaev (CTK), ce qui permettait son décodage. De plus, par observation directe, les excitations du code de couleurs peuvent être identifiées aux défauts de deux copies du CTK, c.-à-d. le groupe de charge est généré par deux paires de semions indépendants. Il semblait alors naturel de se demander s'il existait une transformation de Clifford « simple » (locale) permettant de passer du code de couleurs à deux copies du CTK. Assez rapidement, nous avons construit une telle transformation. La question alors était de savoir si cette transformation se généralisait à tous les codes stabilisateurs topologiques. J'ai tenté de prouver cette affirmation sans succès, alors qu'Héctor y est arrivé. Ma contribution

a donc été de construire des exemples explicites de telles transformations avant que la preuve n'existe pour deux systèmes en particulier, le code de couleurs 4.8.8 et le code de couleurs à sous-systèmes correspondant, puis de les appliquer à leur décodage. J'ai présenté ces résultats lors des conférences annuelles *Southwest Quantum Information and Technology (SQuInT) Workshop 2011*, *APS March Meeting 2011* et *Quantum Information Processing (QIP) 2011*.

5.2 Résumé

Le résultat présenté dans l'article est complexe et ce préambule, qui se veut une introduction pédagogique au résultat, est également relativement complexe. Nous avons démontré explicitement l'équivalence topologique entre différents codes, ce qui nécessite de nombreuses étapes : l'étude des charges topologiques, l'organisation du groupe de charges en fonction de charges élémentaires, les opérateurs de déplacement de charges, et finalement les boucles de déplacement de charges qui deviendront des stabilisateurs. Nous espérons que ce préambule, qui se concentre sur l'exemple concret du code de couleurs 4.8.8 [27], permettra au lecteur de saisir les principales étapes utilisées pour démontrer l'équivalence.

Les travaux présentés dans cet article s'intéressent de manière générale aux codes topologiques stabilisateurs à deux dimensions et invariants sous translation (CTS-IT). Plus précisément, il s'agit des codes stabilisateurs 2D pour lesquels il existe un ensemble de générateurs du stabilisateur qui soit invariant sous translation et où chaque générateur a une taille finie, indépendante de la taille du réseau. Le code de couleurs 4.8.8 et le CTK en sont deux instances. Les sections I et II introduisent les CTS-IT et discutent de la notion d'équivalence de phase. La section III énonce le résultat principal : tous les codes CTS-IT sont équivalents. En termes physiques, cela revient à dire que tous les CTS-IT appartiennent à la même phase topologique. En termes de correction d'erreurs quantique, cela veut plutôt dire que ces codes sont liés par une transformation de Clifford locale. Cette section détaille l'ensemble des étapes qui permettent la preuve du résultat. Ce sont ces différentes étapes que nous illustrons à l'aide de l'exemple traité ci-bas. La section IV discute du cas des codes à sous-systèmes. Finalement, la section V montre comment utiliser le résultat pour décoder des CTS-IT.



(a) Réseau 4.8.8 sur lequel se trouve un qubit par sommet. Nous avons mis en évidence une face f et la frontière d'une autre, $\delta f'$.

(b) Réseau d'arêtes coloriées. La couleur d'une arête est déterminée par la couleur des faces qu'elle relie.

FIGURE 5.1

5.3 Définition du code de couleurs 4.8.8

Considérons un réseau 4.8.8 (cf. Fig. 5.1a). Il est nommé ainsi, car chaque sommet est entouré d'un losange et de deux octogones, qui ont quatre et huit côtés, respectivement. Notons que ce réseau est trivalent, ce qui nous permet d'y définir un code de couleurs. En effet, nous pouvons ainsi munir le réseau d'un coloriage en trois couleurs : les losanges sont rouges, la moitié des octogones verts et l'autre moitié, bleus. Nous en profitons aussi pour colorier les arêtes (cf. Fig. 5.1b) : la couleur d'une arête est donnée par la couleur des deux faces qu'elle rejoint. L'utilité de ce coloriage apparaitra plus loin. Le code stabilisateur se définit comme suit :

1. Un qubit réside sur chaque sommet.
2. Nous définissons deux stabilisateurs sur chaque face f (cf. Fig. 5.1a) :

$$S_X^f = \prod_{q \in \delta f} X_q, \qquad S_Z^f = \prod_{q \in \delta f} Z_q,$$

   où δf est l'ensemble des qubits à la frontière de la face f.
3. Le code est le sous-espace propre +1 de tous les stabilisateurs ou de manière équivalente, le sous-espace fondamental du hamiltonien

$$H = -\sum_f S_X^f - \sum_f S_Z^f.$$

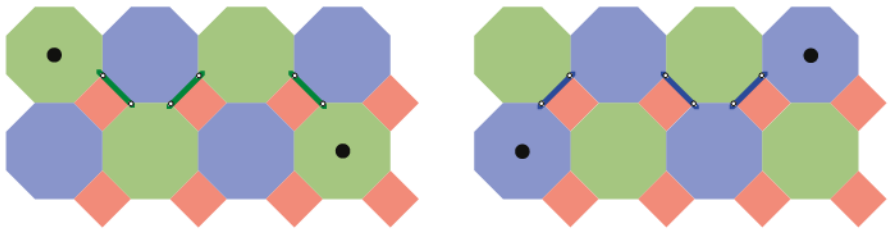
5.4 Charges topologiques

Lorsque le système est dans le code, c.-à-d. dans son état fondamental, et que des erreurs de Pauli se produisent, des excitations apparaissent sur le réseau. L'ensemble de ces excitations forme ce que nous appelons une configuration d'excitations. Les charges topologiques sont définies comme étant les classes d'équivalence des configurations d'excitations : deux configurations d'excitations sont équivalentes s'il existe un opérateur de Pauli au support fini permettant de passer d'une à l'autre. Nous allons montrer que le code de couleurs 4.8.8 exhibe 16 charges topologiques que l'on peut réduire à quatre charges « fondamentales » qui génèrent les 12 autres par composition. Pour ce faire, introduisons les trois sous-réseaux associés aux trois couleurs discutées ci-haut. Nous appellons sous-réseau rouge, l'ensemble des losanges rouges, sous-réseau vert, l'ensemble des octogones verts et sous-réseau bleu, l'ensemble des octogones bleus. Ces sous-réseaux incluent aussi tous les qubits participant aux faces concernées. Dans ce cas-ci, chaque sous-réseau contient tous les qubits du système. Nous définissons aussi les chemins de différentes couleurs. Un chemin d'une couleur donnée est une alternance de faces et d'arêtes de cette couleur, telle que chaque face touche à l'arête suivante et vice-versa.

Dans ce qui suit, nous disons d'une excitation dont le support se réduit à un seul générateur du stabilisateur qu'elle est ponctuelle. De plus, nous disons qu'une excitation ponctuelle est S_X (S_Z) si elle correspond à une valeur propre -1 d'un stabilisateur S_X^f (S_Z^f) pour une face f quelconque. Aussi, nous disons que deux excitations ponctuelles sont du même « type » si celles-ci vivent sur un même sous-réseau et si elles sont toutes deux, soit S_X, soit S_Z. Par exemple, une excitation ponctuelle correspondant à la valeur propre -1 d'un stabilisateur S_X^f où f est un octogone bleu est du type S_X-bleu.

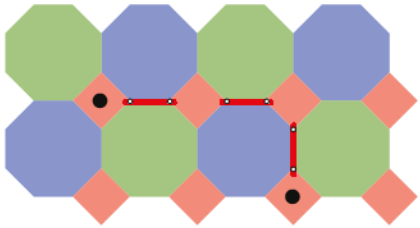
Proposition 1. *Les excitations ponctuelles du même type incluses dans une région connexe peuvent être réduites au vide (à une excitation) par un opérateur de Pauli dont le support est restreint à cette région si leur nombre est pair (impair).*

Démonstration. Considérons deux excitations ponctuelles du même type contenues dans une région R connexe. Choisissons ensuite un chemin γ quelconque dans R de la couleur des excitations et reliant celles-ci. Ce chemin existe toujours, car la région est supposée connexe. Si ces excitations sont S_X (S_Z), nous vérifions directement que l'opérateur $P_Z^\gamma = \prod_{q\in\gamma} Z_q$ ($P_X^\gamma = \prod_{q\in\gamma} X_q$) a pour seul effet de fusionner les deux excitations. La Fig. 5.2 donne un exemple pour chaque couleur de chemins. Les excitations du même type dans R peuvent alors être fusionnées deux à deux par des opérateurs de Pauli n'agissant qu'à l'intérieur de la région d'intérêt R. □



(a) Deux défauts verts aux extrémités d'un chemin vert.

(b) Deux défauts bleus aux extrémités d'un chemin bleu.



(c) Deux défauts rouges aux extrémités d'un chemin rouge.

FIGURE 5.2 Exemples de chemins de chacune des trois couleurs mis en évidence par les traits gras. En appliquant des Z (X) sur les qubits le long du chemin considéré, des défauts S_X (S_Z) de la même couleur sont créés/annihilés.

Proposition 2. *Une excitation ponctuelle S_X-rouge (S_Z-rouge) est équivalente à une paire d'excitations ponctuelles, une S_X-bleue (S_Z-bleue) et une S_X-verte (S_Z-verte).*

Démonstration. Imaginons une région ne contenant qu'une excitation ponctuelle S_X-rouge, c'est-à-dire vivant sur un losange, comme le montre la Fig. 5.3. L'application d'un opérateur Z sur n'importe lequel des sommets du losange transforme cette excitation en une paire d'excitations (S_X-vert, S_X-bleu) sur les octogones adjacents au dit sommet. Un raisonnement similaire s'applique aux excitations du type S_Z-rouge. □

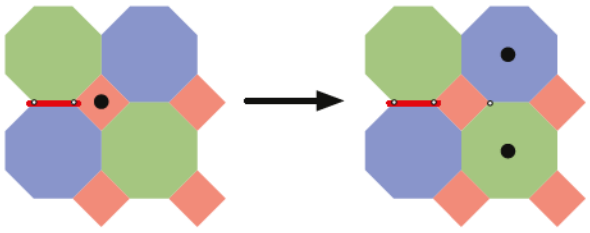


FIGURE 5.3 Un Z (X) sur un qubit du losange (le qubit de droite ici) transforme un défaut S_X-rouge (S_Z-rouge) en une paire des défauts S_X-vert, S_X-bleu (S_Z-vert, S_Z-bleu).

À l'aide des propositions 1 et 2, nous pouvons facilement montrer que toutes les configurations d'excitations peuvent être classées en 16 charges topologiques.

Proposition 3. *Le code de couleurs 4.8.8 possède 16 charges topologiques distinctes. De plus, la charge d'une région est caractérisée par la parité du nombre d'excitations de chacun des types qu'elle contient.*

Démonstration. Étant donné une région et une configuration d'excitations qu'elle contient, nous notons N_X^r, le nombre d'excitations ponctuelles S_X-rouge. De manière similaire, définissons N_Z^r, N_X^v,N_Z^v,N_X^b et N_Z^b. Par la proposition 2, débarrassons-nous de toutes les excitations rouges. Nous avons alors $N'^{\beta}_{\alpha} \leftarrow N_\alpha^\beta + N_\alpha^r$ pour tout $\alpha \in \{X, Z\}$ et $\beta \in \{b, v\}$. Par la proposition 1, réduisons le nombre d'excitations de chacun des quatres types restants à 0 ou 1. Ces nombres d'excitations résiduelles sont donnés par les parités :

$$\pi(N'^{\beta}_{\alpha}) = \pi(N_\alpha^\beta + N_\alpha^r) = \pi(N_\alpha^\beta) \oplus \pi(N_\alpha^r), \tag{5.1}$$

où $\alpha \in \{X, Z\}$ et $\beta \in \{b, v\}$ et où π est l'opérateur de parité et $\oplus$, la somme binaire. □

La preuve met en évidence que les « types » d'excitations ponctuelles étiquettent en réalité différentes charges topologiques du code. Inspirés par la preuve précédente, nous choisissons S_X-bleu, S_Z-bleu, S_X-vert et S_Z-vert comme charges élémentaires. En effet, comme cela est fait dans la preuve, nous pouvons toujours réduire une configuration d'excitations d'une région à un produit de charges élémentaires ponctuelles. Dorénavant, par soucis de concision, nous notons les charges élémentaires c_α^β avec $\alpha \in \{X, Z\}$ et $\beta \in \{b, v\}$. Le groupe de charge $\mathcal{C}$ est le groupe généré par la composition des différentes charges élémentaires

$$\mathcal{C} = \langle c_X^b, c_Z^b, c_X^v, c_Z^v \rangle. \tag{5.2}$$

Tout comme cela est fait à la section 3.1.1 de l'article, il est préférable de faire une granulation (*coarse-graining*) du réseau pour qu'ainsi chaque charge topologique soit représentée sur chaque site du réseau granulé. La façon la plus simple d'y parvenir est de découper le réseau initial en régions contenant deux faces octogones-verts, deux faces octogones-bleus et quatre faces losanges (cf. Fig. 5.4). Sur le réseau granulé, nous ne faisons plus la distinction « sites contenant des qubits » et « faces contenant des générateurs du stabilisateur ». Plutôt, nous rattachons tout aux sites. Comme le support d'un stabilisateur couvre plusieurs sites en général, nous les rattachons aux sites de manière arbitraire, mais systématique. La Fig. 5.4 donne un exemple d'un tel étiquetage. Avec ce choix de granulation, le réseau 4.8.8 devient un réseau carré de sites à 16 qubits et chacun peut contenir au moins une excitation

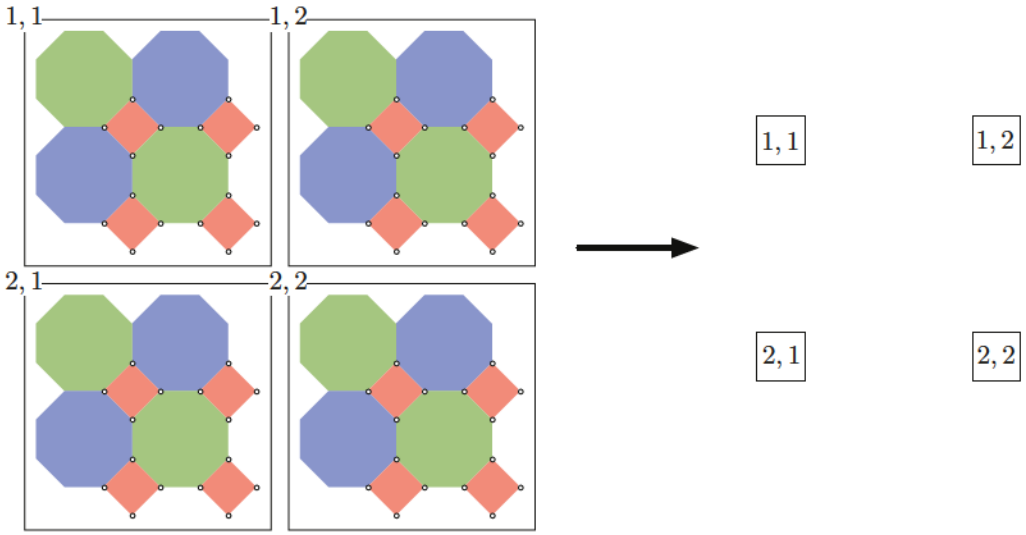


FIGURE 5.4 Granulation du réseau 4.8.8 pour en faire un réseau carré. Chaque site granulé contient 16 qubits. Tous les générateurs du stabilisateur ont une dimension linéaire d'au plus deux sur le nouveau réseau. La figure montre comment rattacher chaque générateur du stabilisateur à un seul site.

ponctuelle de chaque charge élémentaire. De plus, chaque stabilisateur a une taille linéaire d'au plus deux sur le réseau granulé.

Ce n'est pas absolument nécessaire de le faire à cette étape-ci, mais dans le but de simplifier les discussions à venir, nous redéfinissons les générateurs du stabilisateur pour qu'un site ne contienne qu'un générateur par charge élémentaire et que tous les autres soient de charge triviale. Pour ce faire, nous avons besoin de la proposition 4.

Proposition 4. *Considérons deux générateurs du stabilisateur S_1 et S_2 de charges c_1 et c_2 respectivement. Si nous remplaçons le générateur S_1 par $S_1' = S_1 S_2$, alors S_1' a une charge c_1 et S_2 a une charge $c_2' = c_1 c_2$.*

Plutôt que de la démontrer, nous étudions un exemple d'application de la proposition. Prenons les deux générateurs S_X^A et S_X^F de la Fig. 5.5. Définissons $S_X'^F = S_X^A S_X^F$. Puis, remplaçons la paire de générateurs S_X^A, S_X^F par la paire S_X^A, $S_X'^F$. Le groupe généré reste inchangé, car nous pouvons facilement retrouver S_X^F à partir de S_X^A et $S_X'^F$. Voyons maintenant quelle est leur charge respective. Un opérateur n'anti-commutant qu'avec S_X^F dans l'ancien générateur, n'anti-commutera qu'avec $S_X'^F$ dans le nouveau générateur. Nous en déduisons que $S_X'^F$ a la même charge que celle de S_X^F dans l'ancien générateur, c.-à-d. c_X^v. Par contre, S_X^A a maintenant une charge triviale. En effet, comme S_X^A et S_X^F avaient la même charge dans l'ancien générateur, alors il existe un opérateur de Pauli local P n'anti-commutant qu'avec ces deux

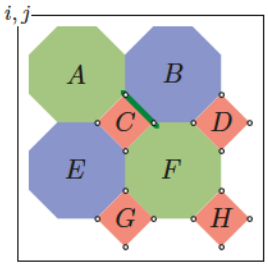


FIGURE 5.5 Un site du réseau granulé qui contient 16 qubits et auquel est associé huit générateurs X du stabilisateur ainsi que huit générateurs Z. L'opérateur P obtenu en appliquant des Z aux deux qubits du trait vert foncé n'anti-commute qu'avec S_X^A et S_X^F.

opérateurs. Il est illustré à la Fig. 5.5 par le trait vert foncé. Or, il s'ensuit que P commute avec $S_X'^F$. Donc, dans le nouveau générateur, P n'anti-commute qu'avec S_X^A. Ce dernier a une charge triviale, car son excitation peut être créée ou annihilée localement. Il est très important de remarquer que la charge d'un stabilisateur dépend du choix de l'ensemble générateur.

Pour n'avoir qu'un générateur par charge élémentaire c_α^β sur un site donné, il suffit de choisir de manière arbitraire un générateur de cette charge et de le remplacer par le produit de tous les générateurs dudit site dont la charge contient c_α^β. Dans le cas qui nous intéresse, nous devons remplacer quatre générateurs. Explicitons le cas de la charge c_X^v en exemple. Reprenons le générateur S_X^F, cf. Fig. 5.5. Remplaçons-le plutôt par le produit $S_X^F \to S_X'^F = S_X^A S_X^F S_X^C S_X^D S_X^G S_X^H$. En effet, avec notre choix de charges élémentaires, S_X^C a une charge composite $c_X^v c_X^b$ et contient donc c_X^v. Il en va de même pour S_X^D, S_X^G et S_X^H. Après ce premier changement, d'après la proposition 4, $S_X'^F$ a une charge c_X^v, S_X^A a une charge triviale et les générateurs des losanges rouges ont une charge c_X^b, car $(c_x^v)^2 = \mathbb{1}$. Puis, nous faisons la même chose pour la charge c_X^b, en substituant $S_X^E \to S_X'^E = S_X^B S_X^E S_X^C S_X^D S_X^G S_X^H$. Il s'ensuit que les générateurs S_X des losanges ont une charge triviale. Nous faisons un traitement similaire pour c_Z^v et c_Z^b. Au final, il n'y a que quatre générateurs sur chaque site dont la charge n'est pas triviale. Chacun a une charge élémentaire différente. Le changement est résumé au Tab. 5.1.

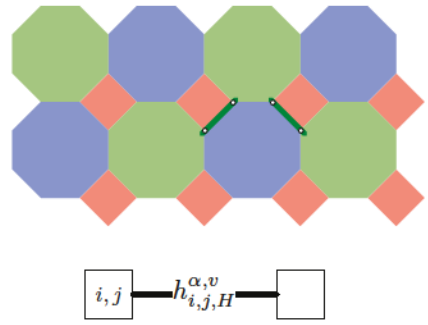
Anciens générateurs	Charge	Nouveaux générateurs	Charge
S_X^A	c_X^v	S_X^A	1
S_X^B	c_X^b	S_X^B	1
S_X^C	$c_X^b c_X^v$	S_X^C	1
S_X^D	$c_X^b c_X^v$	S_X^D	1
S_X^E	c_X^b	$S_X^B S_X^E S_X^C S_X^D S_X^G S_X^H$	c_X^b
S_X^F	c_X^v	$S_X^A S_X^F S_X^C S_X^D S_X^G S_X^H$	c_X^v
S_X^G	$c_X^b c_X^v$	S_X^G	1
S_X^H	$c_X^b c_X^v$	S_X^H	1

Tableau 5.1 Anciens et nouveaux générateurs de type X avec leur charge respective. Un tableau analogue existe pour les stabilisateurs Z.

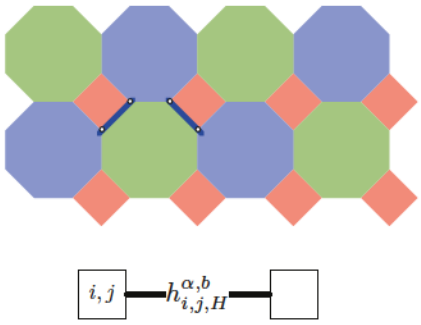
5.5 Opérateurs de saut et de ligne

Les opérateurs de saut sont des opérateurs de Pauli qui créent les mêmes charges sur deux sites adjacents tout en n'excitant aucun autre stabilisateur (de charge triviale ou non). En d'autres termes, ce sont des opérateurs qui permettent aux différentes charges de « sauter » d'un site à l'autre. Pour chaque charge élémentaire un opérateur de saut vertical et un autre de saut horizontal sont définis. Nous notons $h^c_{i,j,H}$ ($h^c_{i,j,V}$) les opérateurs de saut horizontal (vertical) de la charge élémentaire c entre les sites (i,j) et $(i+1,j)$ $\Big((i,j+1)\Big)$. Les opérateurs de saut pour chacune des quatre charges élémentaires sont représentés à la Fig. 5.6.

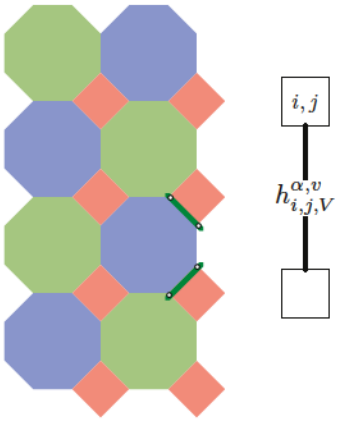
Une fois les opérateurs de saut définis, il est possible de construire des opérateurs lignes le long d'un chemin γ du réseau granulé déplaçant de l'une de ces extrémités à l'autre une charge c. En effet, étant donnée une charge élémentaire c et un chemin γ, l'opérateur $P^c_\gamma = \prod_{(i,j)\in\gamma} h^c_{i,j}$ est cet opérateur ligne (les indices H et V sont omis de la définition, mais sont importants). Pour une charge composite, il suffit de prendre le produit des opérateurs lignes des charges élémentaires correspondantes le long du même chemin.



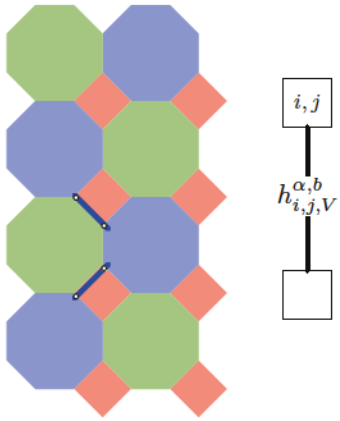
(a) Opérateur de saut horizontal vert.



(b) Opérateur de saut horizontal bleu.

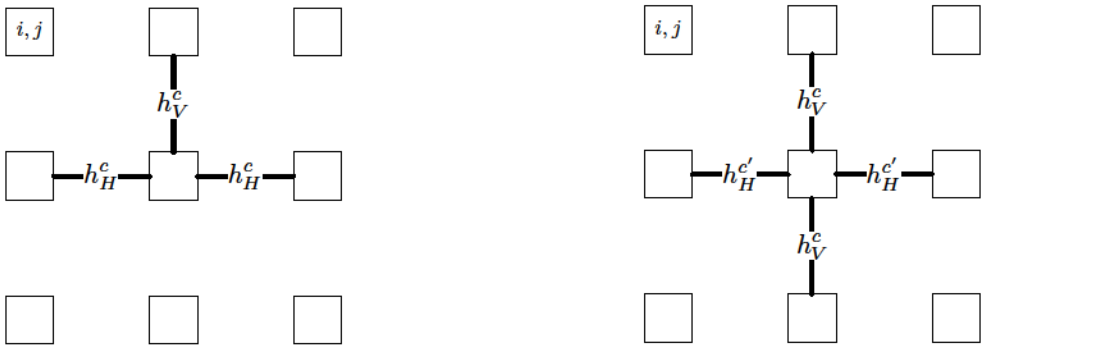


(c) Opérateur de saut vertical vert.



(d) Opérateur de saut vertical bleu.

FIGURE 5.6 Opérateurs de saut $h^{\alpha,\beta}_{i,j}$, où $\alpha \in \{X, Z\}$ et $\beta \in \{b, v\}$: il suffit d'affecter par X ou bien par Z les qubits mis en évidences par les traits gras sur les différentes figures. Ils sont illustrés à la fois sur le réseau 4.8.8 et sur le réseau granulé.



(a) $P_\gamma^c = h_{i+1.j,H}^c h_{i.j+1,V}^c$ et $P_{\gamma'}^c = h_{i+1.j,H}^c h_{i+1,j+1,H}^c$ **(b)** $P_\gamma^c = h_{i.j+1,V}^c h_{i+1.j+1,V}^c$ et $P_{\gamma'}^{c'} = h_{i+1.j,H}^{c'} h_{i+1,j+1,H}^{c'}$

FIGURE 5.7 Opérateurs lignes caractérisant les statistiques topologiques (a) propres et (b) mutuelles.

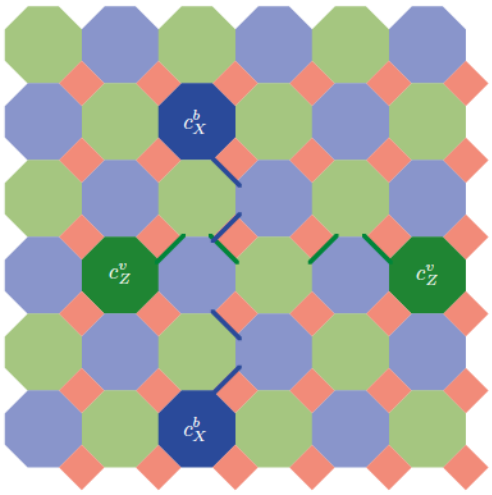
5.6 Statistiques topologiques

À l'aide des opérateurs lignes définis à la section précédente, nous sommes en mesure d'étudier les statistiques topologiques des différentes charges. Comme cela est expliqué à la section 3.1.2 de l'article, ces statistiques peuvent être déduites des relations de commutation des opérateurs lignes définis à la Fig. 5.7 à l'aide de l'expression 5.3,

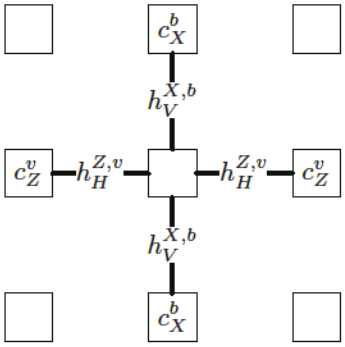
$$P_{\gamma'}^{c'} P_\gamma^c P_{\gamma'}^{c'} P_\gamma^c |\psi\rangle = \pm |\psi\rangle , \tag{5.3}$$

où nous utilisons le fait que les opérateurs lignes sont leurs propres inverses, car ce sont des opérateurs de Pauli. Pour la même raison, seulement deux cas de figure se présentent. Si les opérateurs commutent, $[P_\gamma^c, P_\gamma^{c'}] = 0$, les statistiques, que ce soit propres ou mutuelles, sont triviales, c'est-à-dire bosoniques. Autrement, si $\{P_\gamma^c, P_\gamma^{c'}\} = 0$, les statistiques sont fermioniques ou semioniques, s'il s'agit, respectivement, de statistiques propres ou bien mutuelles. L'avantage de l'équation 5.3 est justement qu'elle nous permette de caractériser des statistiques entre charges différentes, par opposition au test usuel où on échangerait de position des particules indistinguables.

La Fig. 5.8 présente un exemple de statistiques mutuelles non-triviales, à la fois sur le réseau 4.8.8 (Fig. 5.8a) et sur le réseau granulé (Fig. 5.8b). L'anti-commutation des opérateurs lignes est due à l'anti-commutation de $h_H^{Z,v}$ et $h_V^{X,b}$ au site où ils se croisent. Dans le cas du code 4.8.8, nous observons les statistiques suivantes. Toutes les charges élémentaires ont des statistiques propres bosoniques. Aussi, elles ont pour la plupart des statistiques mutuelles bosoniques, à l'exception de deux paires, présentées au Tab. 5.2 qui ont des statistiques mutuelles semioniques. C'est exactement le type d'interaction topologique qui est observé



(a) Sur le réseau 4.8.8



(b) Sur le réseau granulé

FIGURE 5.8 Les statistiques mutuelles de c_X^b et c_X^v sont semioniques, car $\{h_V^{X,b}, h_H^{Z,v}\} = 0$.

$$c_Z^v \leftrightarrow c_X^b$$
$$c_Z^b \leftrightarrow c_X^v$$

Tableau 5.2 Paires de semions pour le code de couleurs 4.8.8.

sur deux copies indépendantes du code topologique de Kitaev contenant chacun une paire de semions.

5.7 Code topologique de Kitaev

Nous supposons une familiarité du lecteur avec le code topologique de Kitaev (CTK). Celui-ci possède deux charges élémentaires. On les note souvent m, la charge magnétique et e, la charge électrique, par analogie aux théories de jauge $\mathbb{Z}_2$ de l'électromagnétisme. Les charges m et e ont des statistiques propres bosoniques et des statistiques mutuelles semioniques. Il s'ensuit que le groupe de charge du CTK est $\mathcal{C}_k = \langle m, e \rangle = \{\mathbb{1}, m, e, f\}$, où f, la particule composite, est un fermion. Dans le cas du code de couleurs 4.8.8, le groupe de charge est généré par les deux paires de semions décrites ci-haut. En notant $\mathcal{C}_{488}$ le groupe de charges du code de couleurs, nous voyons que $\mathcal{C}_{488} \cong \mathcal{C}_K \times \mathcal{C}_K$, où $\times$ dénote le produit

direct de groupes et $\cong$, l'isomorphisme de groupes. Cet isomorphisme, appelons-le Φ, est généré par les relations suivantes :

$$\begin{aligned} \Phi(c_Z^v) &= m_1, & \Phi(c_X^b) &= e_1, \\ \Phi(c_Z^b) &= m_2, & \Phi(c_X^v) &= e_2. \end{aligned} \tag{5.4}$$

Comme une transformation de Clifford locale ne peut modifier le groupe de charges topologiques, il faut compter deux copies du CTK dans notre transformation.

Suivant la construction proposée dans l'article, nous définissons des opérateurs plaquettes du réseau granulé pour chaque charge élémentaire. Nous devons donc en définir quatre types différents. Celles-ci rappellent les opérateurs plaquettes et sites du CTK. Ce n'est pas un hasard. Comme ces plaquettes sont définies de manière invariante sous translation d'un multiple de quatre sites, nous n'en définissons explicitement qu'une seule de chaque type, cf. Fig. 5.9. Le point crucial est de s'assurer que chaque intersection entre opérateurs lignes de charges différentes se produise le long desdits opérateurs, c.-à-d. pas à une des extrémités. En général, cela requiert des plaquettes de taille $(2N+1)\times(2N+1)$, où N représente le nombre de charges élémentaires. Cela est dû au fait qu'en général, les opérateurs lignes peuvent avoir une « épaisseur » de deux sites. Or, dans le cas du code 4.8.8, les opérateurs lignes se limitent vraiment à des lignes, c.-à-d. d'une épaisseur d'un seul site. Cela nous permet de réduire la taille des plaquettes à $(N+1)\times(N+1)$, c.-à-d. 5×5. Comme cela sera utile au moment de spécifier la transformation, nous définissons les opérateurs de saut à cinq sites suivants, cf. Eqs. (5.5)-(5.8), où nous renommons aussi les charges : $c_Z^v \to m_1, c_X^b \to e_1, c_Z^b \to m_2$ et $c_X^v \to e_2$ en accord avec notre isomorphisme Eq. (5.4).

$$\tilde{h}^{m_1}_{I,J,H} = \prod_{j=0}^{3} h^{Z,v}_{4I,4J+j,H} \qquad \tilde{h}^{m_1}_{I,J,V} = \prod_{i=0}^{3} h^{Z,v}_{4I+i,4J,V} \tag{5.5}$$

$$\tilde{h}^{m_2}_{I,J,H} = \prod_{j=0}^{3} h^{Z,b}_{4I+1,4J+j+1,V} \qquad \tilde{h}^{m_2}_{I,J,V} = \prod_{i=0}^{3} h^{Z,b}_{4I+i+1,4J+1,V} \tag{5.6}$$

$$\tilde{h}^{e_1}_{I,J,H} = \prod_{j=0}^{3} h^{X,b}_{4I+2,4J+j+2,H} \qquad \tilde{h}^{e_1}_{I,J,V} = \prod_{i=0}^{3} h^{X,b}_{4I+i+2,4J+2,V} \tag{5.7}$$

$$\tilde{h}^{e_2}_{I,J,H} = \prod_{j=0}^{3} h^{X,v}_{4I+3,4J+j+3,H} \qquad \tilde{h}^{e_2}_{I,J,V} = \prod_{i=0}^{3} h^{X,v}_{4I+i+3,4J+3,V} \tag{5.8}$$

Avec ces opérateurs de saut à cinq sites, nous définissons les plaquettes aux Eqs. (5.9)-(5.10).

$$\pi^{m_1}_{I,J} = \tilde{h}^{e_1}_{I,J,H}\tilde{h}^{e_1}_{I+1,J,H}\tilde{h}^{e_1}_{I,J,V}\tilde{h}^{e_1}_{I,J+1,V} \qquad \pi^{e_1}_{I,J} = \tilde{h}^{m_1}_{I,J,H}\tilde{h}^{m_1}_{I+1,J,H}\tilde{h}^{m_1}_{I,J,V}\tilde{h}^{m_1}_{I,J+1,V} \tag{5.9}$$

$$\pi^{m_2}_{I,J} = \tilde{h}^{e_2}_{I,J,H}\tilde{h}^{e_2}_{I+1,J,H}\tilde{h}^{e_2}_{I,J,V}\tilde{h}^{e_2}_{I,J+1,V} \qquad \pi^{e_2}_{I,J} = \tilde{h}^{m_2}_{I,J,H}\tilde{h}^{m_2}_{I+1,J,H}\tilde{h}^{m_2}_{I,J,V}\tilde{h}^{m_2}_{I,J+1,V} \tag{5.10}$$

Par construction, les opérateurs plaquettes π^c sont des boucles qui ne créent aucune excitation. Ce sont donc des stabilisateurs. De plus, ils sont de charge c. En effet, ils font intervenir des opérateurs de saut de charge $\overline{c}$, la charge opposée à c. Considérons alors un opérateur ligne de charge c' qui possède une extrémité à l'intérieur de π^c et l'autre suffisamment éloignée. Alors, cet opérateur anti-commute avec la plaquette à condition qu'il soit de charge opposée à $\overline{c}$ et commute sinon. En d'autres mots, la plaquette π^c a une charge opposée à la charge opposée à c, c.-à-d. c elle-même. Rappelons que cela est exactement ce qui se produit dans le CTK. Par exemple, une plaquette de charge magnétique est composée d'opérateurs Z, alors que Z est l'opérateur de saut des particules électriques. Pour ajouter ces nouveaux stabilisateurs à l'ensemble générateur, il faut les substituer à d'autres générateurs de même charge. Cela peut être accompli de plusieurs manières. Nous choisissons de remplacer les stabilisateurs énumérés à l'Eq. (5.11) par les plaquettes correspondantes. Ce changement est fait sur tout le réseau de manière invariante sous translation d'un multiple de quatre sites.

$$\begin{aligned}
(S^v_Z)_{2,2} &\leftrightarrow \pi^{m_1}_{0,0} \\
(S^b_Z)_{3,3} &\leftrightarrow \pi^{m_2}_{0,0} \\
(S^b_X)_{4,4} &\leftrightarrow \pi^{e_1}_{0,0} \\
(S^v_X)_{5,5} &\leftrightarrow \pi^{e_2}_{0,0}
\end{aligned} \tag{5.11}$$

Ce faisant, la charge de tous les générateurs inchangés devient triviale. En effet, tous les autres générateurs sont contenus à l'intérieur d'une plaquette correspondant à sa charge, car les plaquettes couvrent l'ensemble du réseau. Pour chaque générateur inchangé, nous pouvons construire un opérateur ligne allant d'un stabilisateur mis de coté à l'Eq. (5.11) jusqu'à lui. La Fig. 5.10 fournit un exemple. L'opérateur ligne en question est local et n'anti-commute qu'avec le générateur considéré. Nous pouvons donc conclure que la charge de ce dernier est triviale. Insistons sur le fait qu'au terme de cette construction, seulement les générateurs plaquettes sont chargés. Tous les autres ont une charge triviale.

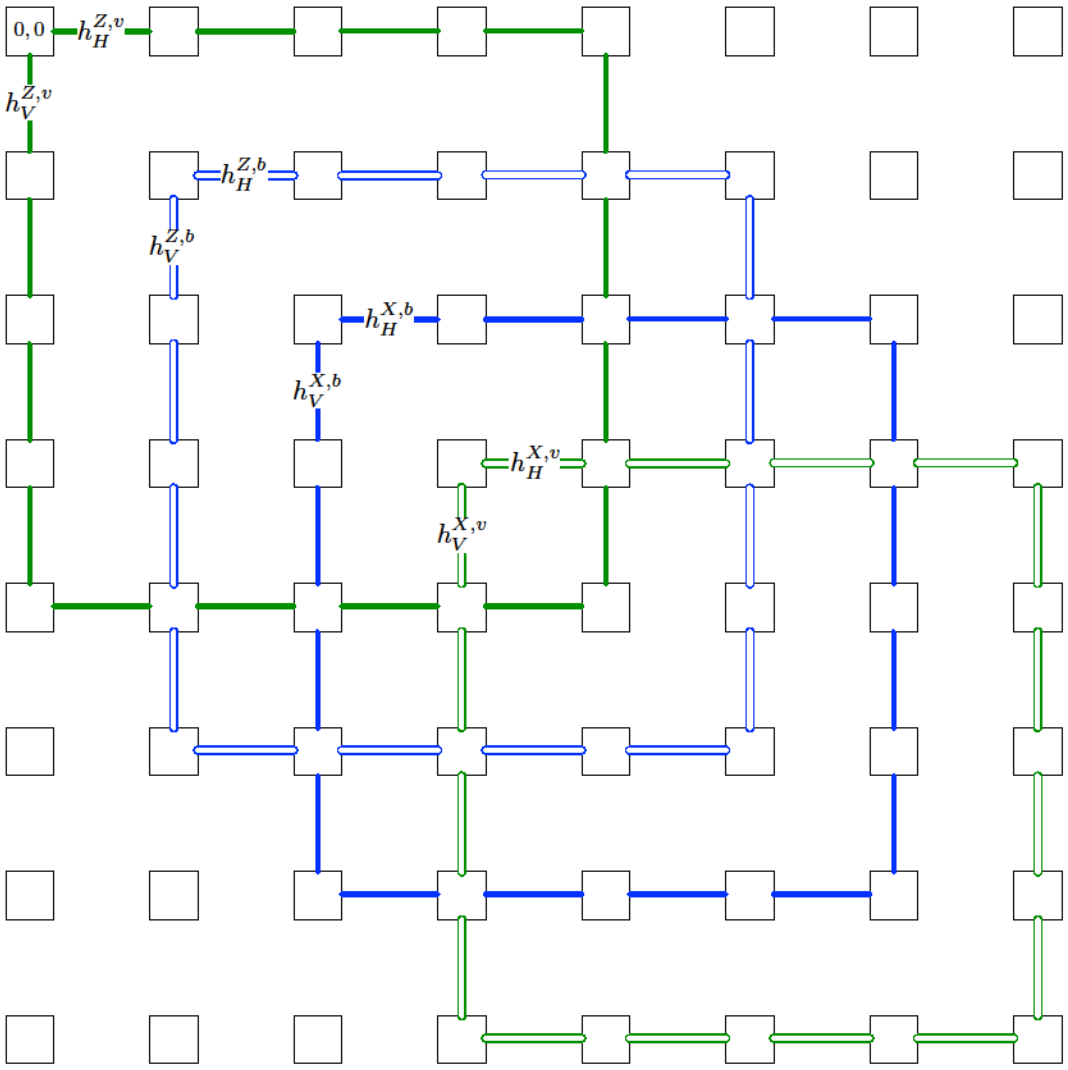


FIGURE 5.9 Un exemple de stabilisateurs plaquettes pour chacune des quatre charges élémentaires. Une translation d'une plaquette par un multiple de quatre sites est aussi une plaquette de même charge.

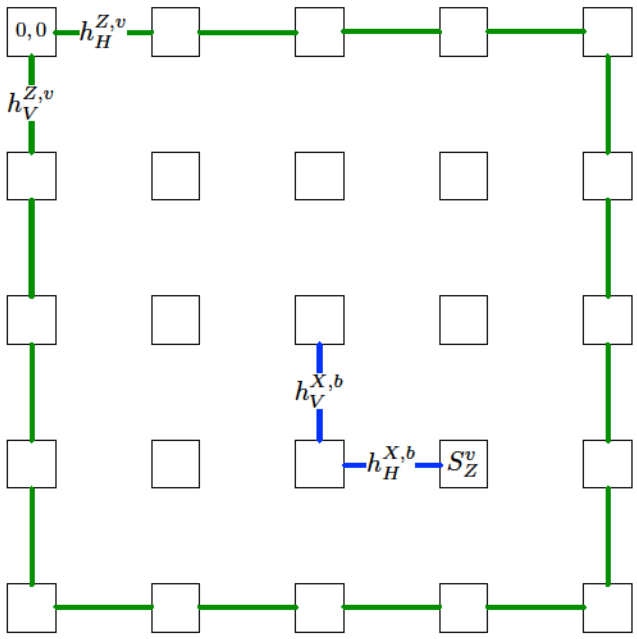


FIGURE 5.10 Un exemple de générateur dont la charge devient triviale : l'opérateur $(S_Z^v)_{3,3}$ a une charge triviale, car l'opérateur ligne $h_{2,2,V}^{X,b} h_{3,2,H}^{X,b}$ n'anti-commute qu'avec lui. En effet, rappelons que $(S_Z^v)_{2,2}$ n'est plus un générateur, cf. Eq. (5.11).

5.8 Transformation de Clifford locale

À l'aide de la définition des plaquettes donnée aux Eqs. (5.5)-(5.8) et (5.9)-(5.10), nous sommes en mesure de donner explicitement la transformation de Clifford locale permettant de passer du code de couleurs 4.8.8 à deux copies du CTK accompagnées de plusieurs qubits désenchevêtrés, correspondant aux stabilisateurs à charge triviale. Dans ce qui suit, nous notons K_1 et K_2 les deux copies du CTK. Rappelons qu'une transformation de Clifford est un automorphisme du groupe de Pauli et peut donc être spécifiée par l'image d'un ensemble générateur du groupe. Nous choisissons ce générateur de telle sorte que les images soient simples. Premièrement, chacun des opérateurs de saut à cinq sites $\tilde{h}_{I,J}^c$ est transformé en opérateur à un qubit.

$$\tilde{h}_{I,J}^{e_1} \to Z_{1,I,J} \qquad \tilde{h}_{I,J}^{m_1} \to X_{1,I,J} \tag{5.12}$$

$$\tilde{h}_{I,J}^{e_2} \to Z_{2,I,J} \qquad \tilde{h}_{I,J}^{m_2} \to X_{2,I,J} \tag{5.13}$$

Par conséquent, toutes les plaquettes $\pi_{I,J}^c$ du réseau granulé sont transformées en opérateurs plaquettes et sites des codes de Kitaev. Comme chaque $\tilde{h}_{I,J}^c$ participe à deux plaquettes et que celles-ci ont quatre côtés, il y a deux fois plus d'opérateurs de saut que de plaquettes. Ceci

est attendu, car les opérateurs de saut servent à la fois à former les opérateurs plaquettes et à produire leur opérateur conjugué. C'est également ce qui est observé dans le CTK. Chaque plaquette contient 16 sites et chaque site contient 16 stabilisateurs. Quatre de ceux-ci ont été remplacés par les plaquettes. Il reste donc $16 \times 16 - 4 = 252$ stabilisateurs à charge triviale par plaquette. Chacun est transformé en un opérateur Z sur un qubit et les opérateurs conjugués à ces stabilisateurs sont transformés en X sur ce même qubit, cf. Fig. 5.10. Ils sont donc désenchevêtrés. En somme, une fraction $4/256$ des qubits sont transformés en deux copies du CTK. La fraction restante, représentant $252/256$ des qubits, est désenchvêtrée. À ce sujet, l'article est trompeur, car la transformation de la figure 12 ne correspond pas à celle présentée ci-haut. Elle a plutôt été trouvée « à la main », tirant profit de la simplicité et de l'abondance de symétries du code de couleurs 4.8.8. C'est pourquoi, mise à part sa qualité pédagogique, nous avons jugé pertinent d'étayer cet exemple. Toutefois, insistons sur le fait que cette construction n'a été conçue avec aucun soucis d'optimalité.

5.9 Article

Universal topological phase of 2D stabilizer codes

H. Bombin,[1] Guillaume Duclos-Cianci,[2] and David Poulin[2]

[1] *Perimeter Institute for Theoretical Physics, 31 Caroline St. N., Waterloo, ON, N2L 2Y5, Canada*
[2] *Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1, Canada*

(Dated: July 30, 2012)

Topological phases can be defined in terms of local equivalence: two systems are in the same topological phase if it is possible to transform one into the other by a local reorganization of its degrees of freedom. The classification of topological phases therefore amounts to the classification of long-range entanglement. Such local transformation could result, for instance, from the adiabatic continuation of one system's Hamiltonian to the other. Here, we use this definition to study the topological phase of translationally-invariant stabilizer codes in two spatial dimensions, and show that they all belong to one universal phase. We do this by constructing an explicit mapping from any such code to a number of copies of Kitaev's code. Some of our results extend to some 2D subsystem codes, including topological subsystem codes. Error correction benefits from the corresponding local mappings. In particular, it enables us to use decoding algorithm developed for Kitaev's code to decode any 2D stabilizer code and subsystem code.

PACS numbers: 03.65.Vf,03.67.Pp

I. INTRODUCTION

The theory of Ginzburg and Landau has had a tremendous success at classifying the different phases of matter in terms of local order parameters and spontaneously broken symmetries. However, it fails to classify certain states of nature, such as the different fractional quantum Hall fluids which all have the same local symmetries. The Hamiltonian of these systems has a constant energy gap, and the ground state degeneracy depends on the topology of the space. Crucially, all ground states are locally identical, which explains the failure of the Ginzburg-Landau paradigm. Instead, the classification of these systems requires the concept of topological order.

Because topological order reflects the long-scale many-body correlations of the system, it cannot be modified locally. This robustness [1–4] is indeed one of the many features that makes topologically ordered systems interesting for quantum information processing [5]. It also suggests a natural classification of topological phases: systems that only differ by a local rearrangement of their degrees of freedom belong to the same topological phase. In other words, the different phases are characterized only by their long-range entanglement patterns [6].

Another, more conventional, description of these phases is in terms of adiabatic connections. If two local and gapped Hamiltonians are connected by a family of local and gapped Hamiltonians, then it should be possible to adiabatically interpolate between the two without encountering a phase transition. The two systems should therefore be in the same phase. This adiabatic evolution will generate a local unitary transformation [7], so consequently the two systems will be in the same topological phase according to the definition adopted above.

Quantum error-correcting codes [8] are intimately related to topological order. To protect the information from local errors, information is encoded into the long-range entanglement of the system. A stabilizer code [9] is a special type of quantum code that can be defined as the degenerate ground state of a Hamiltonian on N qubits of the form

$$H = -\sum_a S_a \quad \text{with} \quad [S_a, S_b] = 0 \quad \forall a, b \qquad (1)$$

where the stabilizer operators S_a are Hermitian elements of the Pauli group, i.e. they are constructed from tensor products of the three Pauli matrices σ_x, σ_y, and σ_z and the identity operator I. Stabilizer codes are also *frustration free*, meaning that the S_a do not generate -1 under multiplication, so the ground states of H are $+1$ eigenstates of all stabilizers, i.e., $S_a|\psi\rangle = +|\psi\rangle$ for all a. The S_a form an Abelian group under multiplication, the stabilizer group $\mathcal{S}$. When the qubits are embedded on a regular lattice, the code—or its associated Hamiltonian—is said to be *local* if each operator S_a has support on a region of constant size, independent of the system size. The support of an operator contains those qubits on which it acts nontrivially.

In this article, we are interested in stabilizer codes that (i) are local and translationally invariant (LTI), and (ii) are topological, in the sense that no local operator can recover any encoded information—i.e., they have a macroscopic minimum distance in terms of error correction or they have no local order parameter in terms of many-body physics. If we place our stabilizer in an infinite lattice, this can be formalized as follows.

Definition 1 *A topological stabilizer code (TSC) is a LTI stabilizer $\mathcal{S}$ such that $\mathcal{Z}(\mathcal{S}) \propto \mathcal{S}$.*

The symbol $\mathcal{Z}(\mathcal{S})$ denotes the centralizer of $\mathcal{S}$, the group of Pauli operators (with bounded support) that commute with all the elements of $\mathcal{S}$. Our main result is that the topological phase of any 2D TSC is uniquely determined by its total quantum dimension $D = 2^n$, or equivalently by its topological entanglement entropy $S_{\text{topo}} = n \log 4$ [10, 11]. This follows from the existence

of a local mapping to n copies of Kitaev's topological code (KTC) [2, 12]. We also adapt the result to a class of subsystem stabilizer codes [13, 14].

Many considerations motivate this line of research. Firstly, stabilizer codes provide simple models to study many-body quantum physics because they often admit exact solutions, and at the same time can exhibit complex phenomena such as topological order and anyonic excitations [2, 12, 15]. To our knowledge, this is the first example where the definition of topological order based on local equivalence [6] can be directly applied to a class of models in a rigorous manner. Secondly, in the context of error correction, the local equivalence to KTC enables us to directly extend a number of properties of this code to all 2D TSCs. For instance, thermal instability [16–19], code tradeoffs [20], logical operator geometry [21], and scale invariance [22] all become trivial corollaries of our mapping. In addition, our mapping provides a method to decode any 2D TSC code, while only a handful of special cases previously had solutions [12, 23, 24]. Thirdly, the local mapping can be used to change encoding during a quantum computation. Because the mapping is local, this change will not propagate errors and is therefore fault-tolerant. This allows to put together the features of different codes—such as having transversal Clifford gates [15], lower weight stabilizer generators [2, 12, 25], etc.—and suggests a natural generalization of the notion of transversality for topological codes to include all local gates.

The rest of this article is arranged as follows. Section II introduces basic definitions. Section III states our main result—the local equivalence of all topological stabilizer codes—and presents a detailed construction of the mapping that realizes this local equivalence. The section ends with an illustration of the mapping for topological color codes [15, 25]. The following section describes how the main results extends to a class of subsystem codes [13, 14], and in particular this is illustrated with the topological subsystem color codes [25]. An important application of the main result is developed in Sec. V where we show how any topological stabilizer code (subsystem or subspace) can be efficiently decoded, a result that extends even beyond the realm of applicability of our main result. We conclude with a brief summary in Sec. VI.

II. DEFINITIONS

The notion of locality plays a crucial role here. For an operator X acting on the qubits of a 2D lattice, let us denote by $|X|$ the range of X, defined as the size of the smallest square containing the support of X. With this definition, a Hamiltonian of the form Eq. (1) is *local* if there exists a constant w such that $|S_a| \leq w$ for all a. A translationally invariant unitary transformation U is *local* if there exists a constant v such that $|U^\dagger X U| \leq |X| + v$ for all operator X. Note that this definition is equivalent [26] to the requirement that U be decomposable into a system-size independent sequence of nearest neighbor unitary transformations (and possibly making use of auxiliary qubits). Lastly, we will say that two local stabilizer codes defined by Hamiltonians H and H' Eq. (1), with stabilizer groups $\mathcal{S}$ and $\mathcal{S}'$, are *locally equivalent* if there exists a local unitary U and two *trivial* LTI stabilizer groups $\mathcal{T}$ and $\mathcal{T}'$ such that $U(\mathcal{S} \otimes \mathcal{T})U^\dagger = \mathcal{S}' \otimes \mathcal{T}'$. A trivial stabilizer group is generated only by single-qubit operators. Physically, U takes the ground state of H onto that of H', and adds or removes extra qubits that are completely unentangled. The existence of renormalization group transformations that disentangle some qubits from topological codes [22] shows the necessity of $\mathcal{T}$ and $\mathcal{T}'$ in this definition.

Kitaev's topological code [2, 12] is defined on a 2D square lattice, with one qubit attached to each edge. For each lattice site s, define an operator $A_s = \prod_{e \in E_s} \sigma_z^e$ where E_s denotes the set of edges incident to site s. Similarly, define for each lattice plaquette p (site of the dual lattice) an operator $B_p = \prod_{e \in E_p} \sigma_x^e$ where E_p denotes the set of edges adjacent to plaquette p. The Hamiltonian of the model is

$$H = -\sum_s A_s - \sum_p B_p. \tag{2}$$

The excitations are anyons, gapped, and topologically charged. Indeed, any set of excitations contained in any finite region of the KTC can be reduced by local operations (i.e. acting in that region) to one of four configurations: the vacuum (0) corresponding to no excitations, an electric charge (e) corresponding to a plaquette excitation B_p, a magnetic charge (m) corresponding to a site excitation A_s, and a composite excitation (f) containing both. These four sectors are the topological charges of the model. Excitations with different charges are characterized by different topological interactions or braiding statistics. According to the effect of exchanging two identical charges, electric and magnetic particles are classified as bosons, while the composite particle is a fermion. As for mutual statistics, they are all semionic because braiding any two distinct non-vacuum charges yields a -1 phase. Finally, two charges can merge to form a new charge. The corresponding fusion rules are Abelian and such that $m \times e \to f$ and $\sigma \times \sigma \to 0$ for $\sigma = m, e, f$.

For general models of the form (1) charge is defined analogously. That is, it labels equivalence classes of excited states up to local transformations, with the understanding that excited states are assumed to be common eigenstates of all the Hamiltonian terms. The notion of topological charge is of utmost relevance because local equivalence preserves the anyon model. Indeed, the anyon model can be derived from commutation properties of certain string operators, which are unaffected by unitary conjugation.

III. MAIN RESULT

We assume that 2D TSCs cannot give rise to chiral anyons [37] because the Hamiltonian terms S_a commute with each other [27]. In our framework, the presence of chiral anyons is defined by certain properties of a matrix introduced later, in definition 2. Because this definition relies on concepts developed in this paper, we save it for later. Under the assumption that there are no chiral anyons, our main result is:

Theorem 1 *Every 2D TSC is locally equivalent to a finite number of copies of KTC.*

By n copies of the code, we mean stacking n lattices on top of each other, each with the same Hamiltonian (2). This result implies that equivalence classes are labeled by the total quantum dimension of the code. Just as the singlet (ebit) can be taken as the fundamental unit of bipartite entanglement [28], our main result suggests that KTC can serve as the fundamental unit of long-range entanglement for TSCs. The result also demonstrates how to systematically search for new translationally invariant stabilizer codes in 2D: take any given number of copies of KT and apply any local unitary in a TI way.

We note that translational invariance is crucial to arrive at this characterization, but also the fact that we are not considering any sort of boundary conditions. Indeed, these can give rise to additional structure that could require interactions between different copies of KTC along the boundaries. A clear example is given by models where the translational symmetry of the excitations is smaller than the translational symmetry of the code. This happens, for example, in Wen's version of KTC [29] and in Bravyi's "strange code" [30], and amounts to the possibility of introducing global topological defects, analogous to the localized topological defects described in [31].

A. Construction of the mapping

In this Section, we give a detailed construction of the mapping between any TSC and copies of KTC. This construction is the core of our main theorem. The only details that we will leave out are 1) that given a Hamiltonian of the form Eq. (1), the S_a can be chosen to be independent under multiplication while preserving locality and translational invariance, and 2) any TSC contains a finite number of topological charges. These two propositions are perhaps not so surprising, but their proof are very cumbersome and can be found in [32], so we take them here as assumptions. It is worth noting that the first assumption rules out the existence of loop-like excitations, as those appearing in the Ising model. Indeed, loops satisfy a local conservation rule, as they cannot have endpoints, and these local conditions would imply the existence of constraints for the S_a.

Note that the first proposition generally requires coarse graining the lattice. With further coarse graining, we can also make sure that all stabilizer generators S_a have support on a 2×2 square. Most steps in the construction require coarse graining the lattice. In what follows, the description of every step assumes the lattice resulting of the coarse graining from all previous steps (though this is certainly not optimal).

1. *Topological charges*

We begin by identifying the topological charges of the TSC at hand. Each eigenstate of the system has a set of excitations $\{S_a\}$, the Hamiltonian terms with negative eigenvalue. As in Kitaev's code, topological charges are equivalence classes of such excitation configurations under local operations. In particular, two configurations with sets of excitations $\{S_a\}$ and $\{S_b\}$ are topologically equivalent when there exists a (finite weight) Pauli operator p that anti-commutes with the Hamiltonian terms $\{S_a\}\triangle\{S_b\}$, where $\triangle$ denotes the symmetric difference of sets —we say that p has syndrome $\{S_a\}\triangle\{S_b\}$. Excitations form a group with product $\triangle$, formalizing the notion of fusion rule here. Since in 2D there is always a finite number of topological charges and because Pauli operators square to the identity, the charge group is isomorphic to the direct product $\mathbb{Z}_2^N$ for some integer N, e.g. $N=2$ in KTC.

We now want to simplify the geometrical layout of the topological charges. We can attach a charge to each stabilizer generator S, namely that of the singleton $\{S\}$. Due to translational symmetry and the finiteness of the number of charges, a suitable coarse graining will make the charge TI and guarantee that every charge can be represented by a set of excitations occupying a single site. Indeed, start by coarse graining till the site at the origin contains, in this sense, all possible charges. Then the charges of the stabilizer generators at any other given site are given by a permutation of those at the origin. But the number of possible permutations is finite and thus there must be two sites $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}+L\mathbf{x}\in\mathbb{Z}^{\mathbf{2}}$ separated by a horizontal distance L as in Fig. 1 with the same permutation. Then the charge of any stabilizer generator $S_a^{\boldsymbol{\alpha}}$ and its horizontal translation $S_a^{\boldsymbol{\alpha}+L\mathbf{x}}$ must be the same. The other axis is analogous, and by coarse graining we get the desired result: a new lattice on which topological charges are represented identically at every site.

2. *Hopping operators and strings*

The next step is to construct hopping operators on this uniform lattice. Given two excitation configurations $\{S_a^{\boldsymbol{\alpha}}\}$ and $\{S_b^{\boldsymbol{\alpha}+\mathbf{x}}\}$ on adjacent sites (or on the same site) and with the same charge, we can choose a Pauli operator with syndrome $\{S_a^{\boldsymbol{\alpha}}\}\triangle\{S_b^{\boldsymbol{\alpha}+\mathbf{x}}\}$, see Fig. 2 a). Since there is a finite number of possible choices for $\{S_a^{\boldsymbol{\alpha}}\}$ and $\{S_b^{\boldsymbol{\alpha}+\mathbf{x}}\}$, and two geometries of adjacency—horizontal and vertical—, after a suitable coarse graining there will

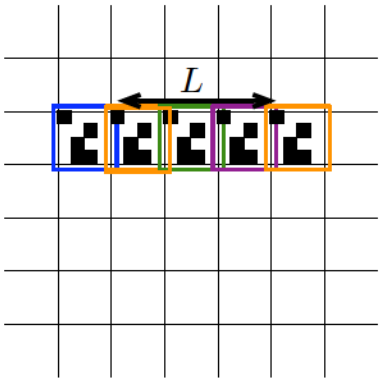


FIG. 1: Black squares in a site represent an excitation configuration. The colored squared around each site represents its charge, different colors corresponding to different charges. The translations of an excitation configuration will in general modify its charge. However, because of translational invariance, it must lead to permutations of the charges. Since the number of charges is finite, one of the permutation must re-

p

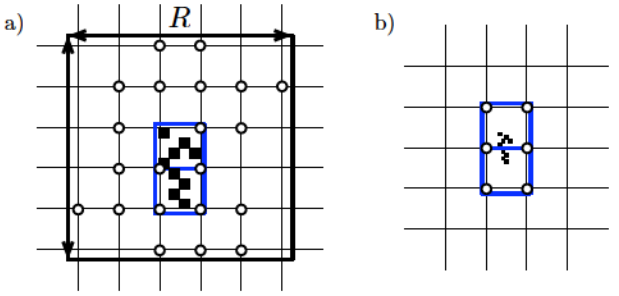


FIG. 2: a) For any two excitation configurations with the same charge on neighboring sites (or on a single site), there must exist a Pauli operator that creates these excitations, by definition of charges. The circles represent the support of that Pauli operator, of range R. b) After coarse graining by the largest such range R, every Pauli operator considered in a) acts only on the direct neighborhood of the two sites (or single site). We call these Pauli operators hopping operators, since they have the effect of moving a topological charge from one site to a neighboring sire (or on the same site).

always be such an operator with support only at these two sites plus those surrounding them, see Fig. 2 b).

Hopping operators can be linked into string operators. Given two excitation configurations $\{S_a^{\boldsymbol{\alpha}}\}$ and $\{S_b^{\boldsymbol{\beta}}\}$, with the same charge c and located on sites $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that form the endpoints of a path γ, there exists a Pauli operator p with syndrome $\{S_a^{\boldsymbol{\alpha}}\}\triangle\{S_b^{\boldsymbol{\beta}}\}$ and with support restricted to the immediate neighborhood of γ, see Fig. 3. We call p a string operator with charge c and endpoints $\{S_a^{\boldsymbol{\alpha}}\}$ and $\{S_b^{\boldsymbol{\beta}}\}$. The string operator only has excitations at its endpoints. Indeed, begin by joining hopping operators of charge c to create the string operator. Consider the site that is at the junction of two such hopping operators. The hopping operators will create two charges c

F
c
o
b

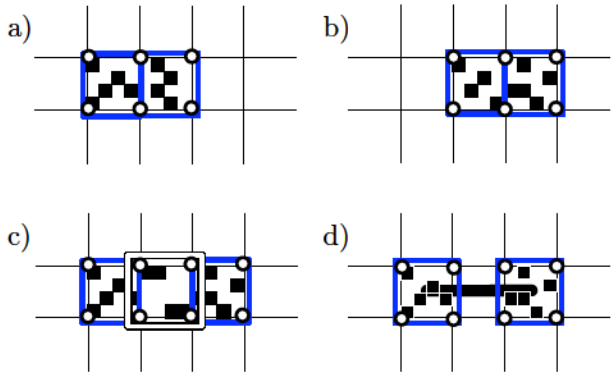


FIG. 4: a) and b) Two hopping operators of same charge overlapping on one plaquette. c) Product of the two operators of a) and b). Because charges square to identity, the middle plaquette contains an excitation of trivial charge. By definition of charge, there exists an operator that can correct for this excitation (and acting only on this plaquette, see text). d) Small string operator resulting of this procedure. Iterating yields arbitrary long string operators with arbitrary configurations (of same charge) on the endpoints.

at that site, resulting in a trivial charge, but may have a non-trivial syndrome. However, by the previous paragraph, there exists a Pauli operator on that site with the resulting syndrome. By including this Pauli operator in the construction of the string operator, we obtain the desired result, see Fig. 4.

Anyonic statistics are recovered from string operators [33]. Namely, mutual statistics of two charges c and c' are trivial [semionic] when two crossing string operators with charges c and c' [anti]commute. Similarly, a given charge c is bosonic [fermionic] if, given three string operators q_i with charge c and with a common endpoint, the operators q_1q_2 and q_1q_3 [anti]commute, see Fig. 5 — three such string operators are enough to represent a process where two identical anyons are exchanged. These commutation properties are independent of the strings chosen, as shown in Fig. 5, so they indeed encode the topological properties of the excitations.

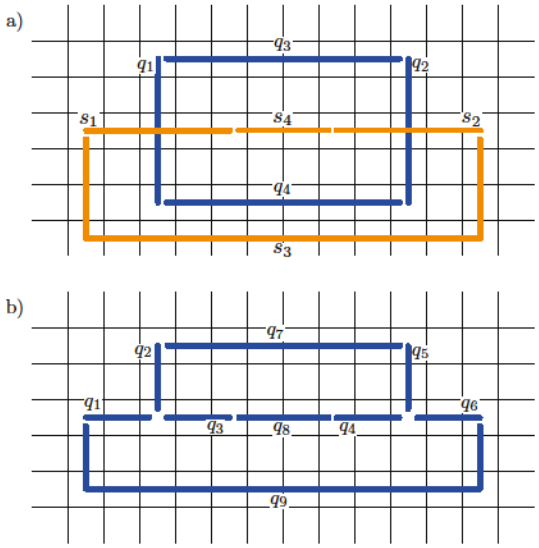


FIG. 5: a) Mutual statistics do not depend on the microscopic details of strings: only their respective charges determine the commutation relations. For two Pauli operators a and b, let $(a,b) = 1$ if a and b commute and $(a,b) = -1$ otherwise. Then, for two different choices of strings q_1, s_1 and q_2, s_2, we must have $(q_1, s_1) = (q_2, s_2)$ because $q_1q_2q_3q_4$ and $s_1s_2s_3s_4$ are stabilizer operators and consequently must commute. b) A similar reasoning holds for self-statistics: $(q_1q_2, q_1q_3) = (q_4q_5, q_4q_6)$ because $q_1q_2q_7q_5q_6q_9$ and $q_1q_3q_8q_4q_6q_9$ are stabilizer operators.

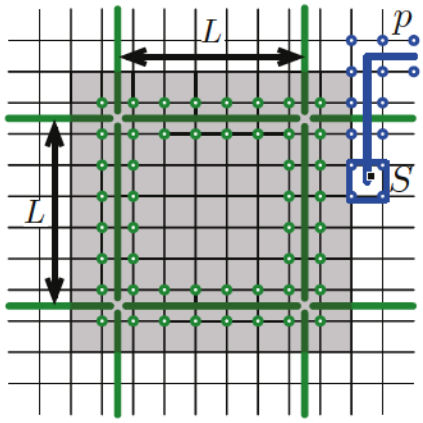


FIG. 6: Lattice of horizontal and vertical segment operators of length L and of a given charge (e.g. green). Stabilizer generators of a plaquette π must be contained in the shaded region. Indeed, if there existed at least one generator outside this region (represented by a black square), then we could build a string operator anti-commuting only with this particular generator (we imagine the other endpoint to be as far as needed from the plaquette). Thus, the string must also anti-commute with the plaquette. However, the plaquette and the string do not share support (green and blue circles respectively) and then must commute, a contradiction.

3. *Plaquettes*

Given a non-trivial charge c and a length L, we can construct a lattice $\Lambda_c = \{\alpha_0 + nL\mathbf{x} + mL\mathbf{y} | n, m \in \mathbb{Z}\}$ of segment operators as in Fig. 6. Each segment is a horizontal or vertical string of charge c with common endpoints at sites separated by a distance L. The product of the segments forming a plaquette π on Λ_c is, up to a phase, a stabilizer. This follows from the fact that it has trivial syndrome and is local, and only stabilizers can have that property by definition of TSC. Thus, such a plaquette operator π must be proportional to a product of stabilizer generators $\{S_a\}$. The support of these $\{S_a\}$ is highly constrained. First, if the support of a stabilizer generator S is not contiguous to the smallest ball containing the plaquette operator π, then S cannot be one of the S_a generating π. To prove this, we can construct a string p as in Fig. 6 with one endpoint S and the other endpoint as far away as needed so that it does not overlap with any of the S_a. This p commutes with π because they do not share support, but anti-commutes only with the stabilizer generator S (and possibly other stabilizer generators arbitrarily far), showing that S cannot be one of the generators of π. Second, there cannot be a hole inside the plaquette π where none of the S_a generating π has support, see Fig. 7. To prove this, we can construct the product q of those S_a with support in the lower half of the lattice, see Fig. 7. The resulting string q is not closed and is a stabilizer, so it must have trivial charge, but it coincides with p in the lower part of the plaquette, a contradiction since p is charged.

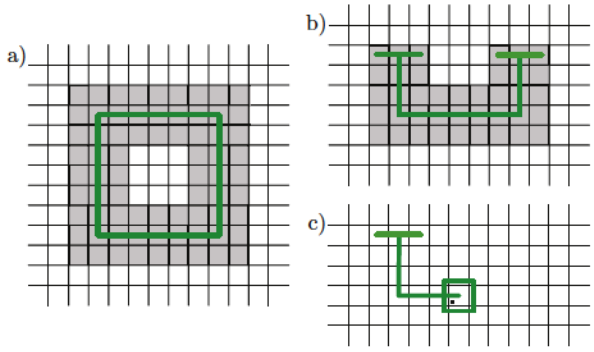


FIG. 7: The shaded regions represent the union of the support of the stabilizer generators $\{S_a\}$ entering into the decomposition of π. The green loop is a string operator forming the plaquette π. a) We consider the possibility that the support of the stabilizer generators of a plaquette has a hole inside it. b) By multiplying the bottom half of the stabilizer generators we obtain a stabilizer operator which locally looks like a green string on its lowest portion. c) If we cut this operator in half vertically, its new endpoint will contain an excitation of green charge, since it is locally identical to the green string. However, the other endpoint has no excitation at all since it is constructed with stabilizers generators, so it holds a trivial charge, a contradiction.

As a consequence of these geometrical constraints, we

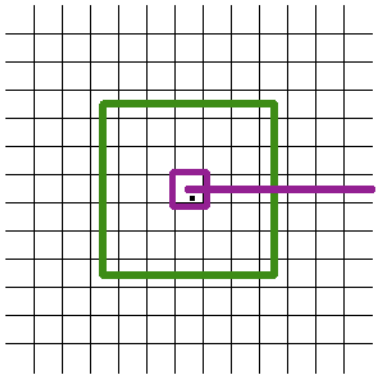


FIG. 8: First, we consider any of the generators, say S_i, of a plaquette, π_i, which is in its center (represented by the black square). We know such an operators exists (cf. Fig. 7). Moreover, this stabilizer generator is not involved in any other plaquette due to its location far from the boundary (cf. Fig. 6). We can then substitute S_i and π_i in our stabilizer generator set and we do this for every plaquette in a TI way. Second, we consider a string operator which anticommutes only with the stabilizer generator S_i (we put the other end arbitrary far). It follows that this string operator anticommutes with π_i which in turn implies that it anticommutes with the string segment it crosses. This means the two charges (green and purple) have semionic interaction.

can choose in a TI way for each plaquette $\pi^{\boldsymbol{\alpha}}$ a stabilizer generator $S^{\boldsymbol{\alpha}}$ on its central area, see Fig. 8, that is not shared with the other plaquettes. We can substitute the stabilizer generator $S^{\boldsymbol{\alpha}}$ by $\pi^{\boldsymbol{\alpha}}$—adjusting the phase so that it is a stabilizer—to get a new set of independent LTI stabilizer generators of $\mathcal{S}$. Moreover, by considering a string p with an endpoint $S^{\boldsymbol{\alpha}}$ and the other one away from $\pi^{\boldsymbol{\alpha}}$, see Fig. 8, we see that c (the charge of the segment operators used to construct π) and the charge of p have semionic mutual statistics.

4. *Canonical charges*

It follows from this last observation that the only charge that braids trivially with all other charges is the trivial charge; for all other charge c there exists at least one charge c' with which it has semionic statistics. Using this fact, we now want to organize topological charges into a canonical form. For doing this, we need to pick a set of N "elementary charges" c_i that generate all other charges under fusion. Consider the $N \times N$ symmetric matrix $\mathbb{S}_{ij}$ over $\mathbf{Z}_2$ whose i-th diagonal element is 0 [1] if c_i is a boson [fermion] and off diagonal element ij is 0 [1] when the mutual statistics of c_i and c_j are trivial [semionic]. This matrix can be used to define the notion of a chiral model, used in the statement of our main theorem.

Definition 2 *A TSC is chiral if* $Tr\mathbb{S} \neq 0$.

We can transform the matrix $\mathbb{S}$ into the canonical form $I_{N/2} \otimes \sigma_x$, assuming that the anyon model is not chiral (in which case the last two diagonal entries would be 1). First, notice that exchanging c_i with c_j has the obvious effect of permuting rows and columns of $\mathbb{S}$ and that substituting c_i with $c_i c_j$ will produce a new matrix with $\mathbb{S}'_{ii} = \mathbb{S}_{ii} + \mathbb{S}_{jj} + \mathbb{S}_{ij}$, $\mathbb{S}'_{ij} = \mathbb{S}_{ij}$ and $\mathbb{S}'_{ik} = \mathbb{S}_{ik} + \mathbb{S}_{jk}$ for $k \neq i, j$ (plus symmetric equations so that $\mathbb{S}$ remains symmetric), while other elements remain the same. As a first step, permute the c_i-s such that $\mathbb{S}_{12} \neq 0$, which is always possible. We then perform a Gaussian elimination. For $i > 2$ perform the substitutions $c_i \longrightarrow c_i c_1^{\mathbb{S}_{2i}} c_2^{\mathbb{S}_{1i}}$ so that $\mathbb{S}_{1i} = \mathbb{S}_{2i} = 0$. Repeating this $N/2-1$ more times we get $\mathbb{S} = I_{N/2} \otimes \sigma_x + \mathbb{D}$ with $\mathbb{D}$ a diagonal matrix with entries $\mathbb{D}_i$. If $\mathbb{D}_{2i-1} = 0$ and $\mathbb{D}_{2i} = 1$ substitute $c_{2i} \longrightarrow c_{2i-1}c_{2i}$, and similarly for $\mathbb{D}_{2i-1} = 1$ and $\mathbb{D}_{2i} = 0$, so that we get $\mathbb{D}_{2i-1} = \mathbb{D}_{2i} = 0$. Then, there must be an even number of indices i such that $\mathbb{D}_{2i-1} = \mathbb{D}_{2i}$ (otherwise the model would be chiral). Pick any pair i, j of such indices and substitute

$$c_{2i-1} \longrightarrow c_{2i-1}c_{2j}, \tag{3}$$
$$c_{2i} \longrightarrow c_{2i-1}c_{2i}c_{2j}, \tag{4}$$
$$c_{2j-1} \longrightarrow c_{2j-1}c_{2i}, \tag{5}$$
$$c_{2j} \longrightarrow c_{2j-1}c_{2j}c_{2i}. \tag{6}$$

Repeating this procedure we arrive finally to $\mathbb{D} = 0$ as desired. Thus, we obtain a set of N canonical generating charges e^i and m^i, $i = 1, \ldots, N/2$, that interact topologically as if they were the electric and magnetic charges of $N/2$ copies of KTC.

Since every site of the lattice contains an excitation of every topological charge, we can naturally identify these canonical elementary charges e^i and m^i with stabilizer operators at a given site, in a TI way. Furthermore, we can change the generators of $\mathcal{S}$ in a TI way to include these stabilizer operators associated to elementary charges. Just notice that if S and $\{S_i\}$ are stabilizer generators with charges c and c_i, substituting each S_i with $S'_i = SS_i$ will give rise to a new set of independent stabilizer generators where S has charge $c\prod_i c_i$ and S'_i has charge c_i. Thus, to every elementary charge c, we can associate stabilizer generators $S_c^{\boldsymbol{\alpha}}$ in a TI way.

Take any canonical electric charge e^i and choose a stabilizer generator S_{e^i} of charge e^i. Construct a TI lattice of segment operators with common endpoints at S_{e^i} and its translations. There are four segments meeting at S_{e^i}, call them q_i, $i = 1, \ldots, 4$, see Fig. 9. If q_i does not commute with q_1, substitute it with $S_{e^i}q_i$, and do this at the other endpoints in a TI way. Thanks to the bosonic character of e_i, the new segment operators all commute with each other. We can adjust their phases so that they are Hermitian. Then, plaquette operators π constructed from the lattice of segment operators are either stabilizer or stabilizers with a negative sign. In the second case, we can negate those horizontal segment operators at every other line, which takes us back to the first case. Now choose one of these plaquette stabilizers π, a stabi-

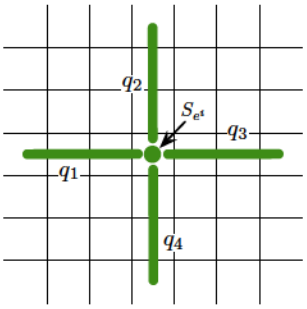


FIG. 9: We consider the meeting points of the segment ope ators of the lattice used to build plaquettes earlier (cf.Fig. (We can ensure that these string operators commute with o another. Take any stabilizer generator, S_{e^i}, that is contain in the meeting point and that has the same charge as the se ments, say e^i. Then if $(q_1, q_2) = -1$, we multiply q_1 by S such that $(S_{e^i} q_1, q_2) = 1$. The bosonic character of e^i ensur that the remaining commutations among the q_i are as desire e.g. $(S_{e^i} q_1 q_2, S_{e^i} q_1 q_3) = 1$ implies $(S_{e^i} q_1, q_3) = 1$.

lizer generator S_{m^i} with charge m^i and a string p wi an endpoint m^i and the other one far away, as in Fig. Then p anti-commutes with π and S_{m^i}, showing that S_{m^i} is one of the generators of π. Then we can proceed above and attach, in a TI way, an exclusive generator S_{m^i} with charge m^i to each plaquette $\pi^{\boldsymbol{\alpha}}$. As above, we c substitute $S^{\boldsymbol{\alpha}}_{m^i}$ with $\pi^{\boldsymbol{\alpha}}$ to get a new set of independe LTI stabilizer generators. Clearly each $\pi^{\boldsymbol{\alpha}}$ has charge m In this argument we can of course exchange e^i and m^i

5. *Canonical stabilizer generators and mapping*

We can put together N such lattices, one per canoi cal charge generator, as in Fig. 10. We refer to the la tice constructed from e^i (m^i) segment operators as t e^i (m^i) lattice. This particular geometry guarantees th the commutation relations of segment operators only d pend on their charge. Notice that a plaquette π in the lattice contains a single vertex of the m^i lattice, so th we can attach to π the corresponding stabilizer generat S_{m^i} with charge m^i that lies at that vertex. Again tl works just as well exchanging e_i and m_i. It follows that we can replace each stabilizer generator that is the end- point of segment operators of one of the N lattices with the enclosing plaquette of the same charge (see Fig. 10), obtaining a new set of independent LTI stabilizer gener- ators.

This new set of stabilizer generators breaks into two disjoint subsets. The first subset of stabilizer generators are the plaquette operators constructed in the previous paragraph. The second subset contains all the other sta- bilizer generators. These generators commute with seg- ment operators and have trivial charge. To prove this second statement, let S be a stabilizer generator of the second kind. Then there exists a number of plaquette op- erators $\{\pi_a\}$ with the same total charge as S and a Pauli

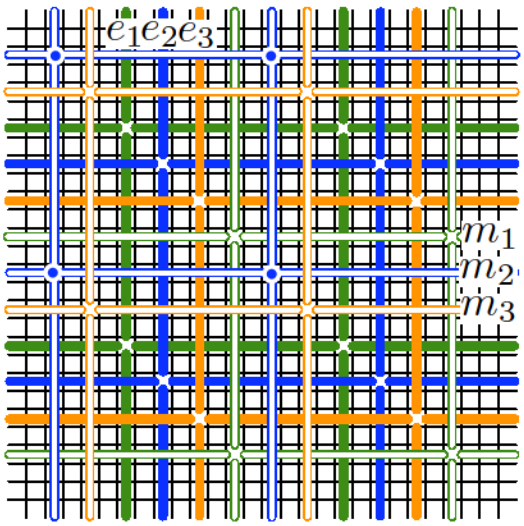


FIG. 10: $2N$ lattices of segment operators, where N is the number of canonical pairs of charges (here $N = 3$). The re- spective offsets are chosen to insure that strings of different charges cross properly (if they do overlap) such that their commutation relations are well defined. A stabilizer gener- ator of each plaquette can be chosen to be replaced in the stabilizer generator set. Here, an example is given for the e^2 lattice (blue): the blue circles represent stabilizer generators of charge S_{m^2}.

operator p with syndrome $\{S\} \cup \{\pi_a\}$ (this is true because plaquette operators generate all charges by construction). But every Pauli operator p anti-commutes with an even number of plaquettes of a given charge. The reason for this is much like in KTC. If p anti-commutes with a given segment operator, then the two plaquette operators that share this segment operator are affected, so excitations of any given elementary charge always appear in pairs. It follows that the charge of $\{\pi_i\}$, and thus of S, is trivial.

The segment operators form a canonical basis for a sub- group $\mathcal{P}_S$ of Pauli operators. Every e^i-segment is crossed by a unique m^i-segment, so they form a canonical pair. In addition, all other pairs of segment operators commute thanks to the canonical form of the $\mathbb{S}$ matrix. Thus, any given Pauli operator p factorizes as $p = p_1 p_2$—uniquely up to a sign—, with p_1 in $\mathcal{P}_S$ and p_2 in its centralizer $\mathcal{Z}(\mathcal{P}_S)$.

For any stabilizer generator S that is not a plaquette formed of segment operators as in Fig. 10, there exists a Pauli operator $\tilde{S}$ that anti-commutes with S and with no other stabilizer generator, simply because the charge of S is trivial. Due to the previous point, we can choose $\tilde{S} \in \mathcal{Z}(\mathcal{P}_S)$ while preserving locality and TI. Indeed, de- compose $\tilde{S} = \tilde{S}_1 \tilde{S}_2$ with $\tilde{S}_1 \in \mathcal{P}_S$ and $\tilde{S}_2 \in \mathcal{Z}(\mathcal{P}_S)$ and substitute $\tilde{S}$ with $\tilde{S}_2$.

Label all the segment operators $\sigma_c^{\boldsymbol{\alpha}}$ by their charge $c = e^i$ or m^i and lattice location $\boldsymbol{\alpha} \in \mathbb{Z}^2$ (lattice sites are now located at the crossing of segment operators, so it is tilted at 45°). Similarly, label all the non-plaquette stabilizer

generators $S_a^{\boldsymbol{\alpha}}$, where a labels the different stabi
a given lattice site. We can arrange the $S_a^{\boldsymbol{\alpha}}$ and
canonical form. Choose an ordering in $\mathbb{Z}^2$ that
that $\boldsymbol{\alpha} < \boldsymbol{\beta}$ iff $\boldsymbol{\alpha}+\boldsymbol{\gamma} < \boldsymbol{\beta}+\boldsymbol{\gamma}$, e.g. $(\alpha_x, \alpha_y) < (\beta$
$\alpha_y < \beta_y$ or $\alpha_y = \beta_y$ and $\alpha_x < \beta_x$. Extend that
on pairs $(\boldsymbol{\alpha}, a)$, for instance $(\boldsymbol{\alpha}, a) < (\boldsymbol{\beta}, b) \Leftrightarrow \boldsymbol{\alpha}$
$\boldsymbol{\alpha} = \boldsymbol{\beta}$ and $a < b$. For each $\tilde{S}_a^{\boldsymbol{\alpha}}$, consider the fini

$$B(\tilde{S}_a^{\boldsymbol{\alpha}}) := \{(\boldsymbol{\beta} \in \mathbb{Z}^2, b) | (\boldsymbol{\beta}, b) < (\boldsymbol{\alpha}, a), [\tilde{S}_a^{\boldsymbol{\alpha}}, \tilde{S}_b^{\boldsymbol{\beta}}$$

and perform the substitution

$$\tilde{S}_a^{\boldsymbol{\alpha}} \leftarrow \tilde{S}_a^{\boldsymbol{\alpha}} \prod_{(\boldsymbol{\beta},b)\in B(\tilde{S}_a^{\boldsymbol{\alpha}})} S_b^{\boldsymbol{\beta}}.$$

Then the $\tilde{S}^{\boldsymbol{\alpha}}$ are TI and together with the $S^{\boldsymbol{\alpha}}$
canonical Pauli basis of $\mathcal{Z}(\mathcal{P}_S)$.

We have thus constructed a local TI canonic
for the Pauli group consisting of pairs of segme
ators $(\sigma_{e^i}^{\boldsymbol{\alpha}}, \sigma_{m^i}^{\boldsymbol{\alpha}})$ and pairs $(S_a^{\boldsymbol{\alpha}}, \tilde{S}_a^{\boldsymbol{\alpha}})$ consisting
lizer generators with trivial charge and their car
conjugated partner. The mapping to KTC is n
ial. The $(\sigma_{e^i}^{\boldsymbol{\alpha}}, \sigma_{m^i}^{\boldsymbol{\alpha}})$ map to the single qubit Pau
tors $(\sigma^{z,\boldsymbol{\alpha},i}, \sigma^{x,\boldsymbol{\alpha},i})$ where i labels the $\frac{N}{2}$ distinc
of KTC and $\boldsymbol{\alpha}$ labels the (tilted) lattice sites.
tain the usual picture of qubits located on edge
lattice by choosing a lattice rotated by 45°. T
tional Pauli operators $(S_a^{\boldsymbol{\alpha}}, \tilde{S}_a^{\boldsymbol{\alpha}})$ are mapped o
iliary qubits $(\sigma^{z,\boldsymbol{\alpha},a}, \sigma^{x,\boldsymbol{\alpha},a})$. The constraint S
implies that these auxiliary qubits are all in t
$|0\rangle$.

B. Example

We illustrate this mapping for topological col
(TCCs) [15, 25]. A TCC can be constructed o
valent lattice with 3-colorable faces, but we take
ticular the square-octagon regular lattice of Fig.
lattice is particularly useful in terms of fault-t
[15]. Qubits are located at the vertices of the lat
there are two stabilizer operators per plaquette

$$S_p^{\sigma} = \bigotimes_{e\in E_p} \sigma^e, \quad \text{with } \sigma \in \{\sigma_x, \sigma_z\}.$$

The excitations in this model carry 16 different
ical charges that correspond exactly to the cha
tained from two copies of KTC. For the two c
KTC, these 16 charges are generated by the fc
mentary" charges e^j and m^j with $j = 1, 2$ labe
two KTCs. Among the 16 charges of the color code, we
can choose four with the same topological interactions as
the e^j, m^j. Furthermore, we can find "hopping opera-
tors" for these elementary charges. In KTC, the hopping
operator for, say, the charge e^1 is a σ_z operator on the
first copy of KTC, as this operator has the effect of mov-
ing an e^1 charge around. Once these elementary hopping

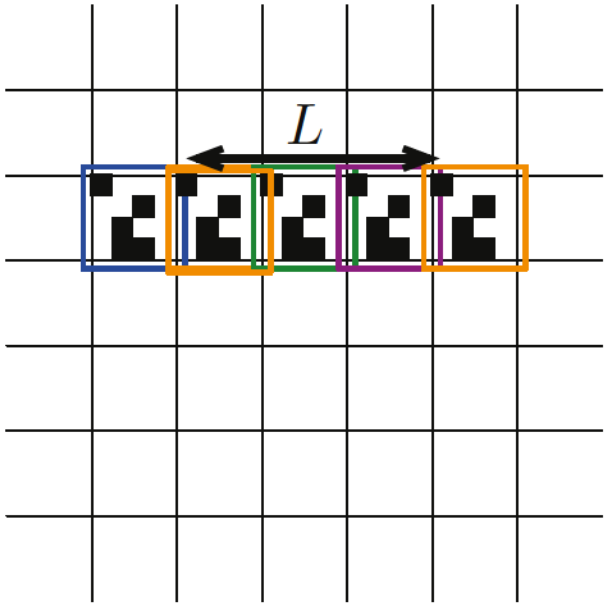


FIG. 11: Regular square-octagon lattice for TCC. The diamonds can be labeled A or B according to a chessboard pattern. There are two stabilizers Eq. (7) associated to each plaquette. Here is an example of the mapping from one TCC to two copies of KTC. The black dots (stars) represent σ_z (σ_x) operators. A Z-plaquette on a A-diamond of the TCC gets mapped to a plaquette operator on the first KTC and to a site operator on the second KTC. The complete mapping for 1-qubit Pauli operators is shown in Fig.12.

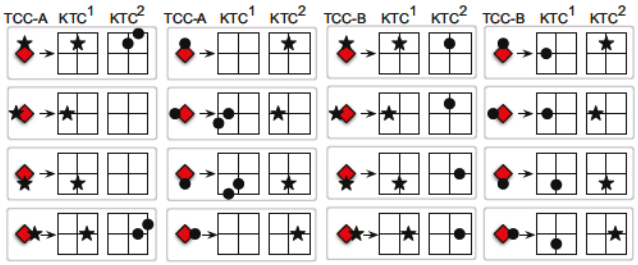


FIG. 12: Mapping between the 1-qubit Pauli operators of the square-octagon TCC an two copies of Kitaev's code KTC^1, KTC^2. The first (last) two columns are for the A (B) sublattice. Circles (stars) represent σ_z (σ_x) operators. For instance, the upper left diagram indicates that a σ_x located at the top of a diamond of the A sub-lattice gets mapped to a σ_x on KTC^1 and two σ_z on KTC^2. All commutation relations are preserved by this mapping, so it is unitary and obviously local.

operators have been identified, the mapping proceeds by identifying the hopping operators of the color code with those of the two KTCs. This procedure leads to the mapping shown at Fig. 12. It can be directly verified that it maps stabilizer generators of TCC to stabilizer generators of two KTCs, in this case with no need to add or remove trivial stabilizers.

IV. EXTENSION TO SUBSYSTEM CODES

Subsystem stabilizer codes form a more general class of stabilizer codes [13, 14]. They can be defined as a pair $(\mathcal{S},\mathcal{G})$, where $\mathcal{G}$ is an arbitrary Pauli subgroup and $\mathcal{S}$ a stabilizer such that $\mathcal{S} \propto \mathcal{Z}(\mathcal{G}) \cap \mathcal{G}$. Encoding is not done on the whole subspace defined by $\mathcal{S}$, but rather on the subsystem where the action of $\mathcal{G}$ is trivial. This way, errors caused by operators in $\mathcal{G}$ do not affect encoded states. Because of this, elements of $\mathcal{G}$ are called gauge operators.

We say that a subsystem code $(\mathcal{S},\mathcal{G})$ is LTI if $\mathcal{G}$ admits a LTI set of generators $\mathcal{G}_b$. Note that some local subsystem codes admit no local stabilizer generators, e.g. [14]. Unlike them, a topological subsystem code should have a stabilizer with a local description. In addition, local operators should not recover any encoded information. Since we do not care about the effect of gauge operators, this can be formalized as follows in an infinite lattice:

Definition 3 *A topological stabilizer subsystem code (TSSC) is a LTI subsystem stabilizer code* $(\mathcal{S},\mathcal{G})$ *such that* $\mathcal{Z}(\mathcal{S}) \propto \mathcal{G}$.

There is a general strategy to understand TSSCs in terms of TSCs. Namely, to find a TSC $\mathcal{S}'$ that lies in between the stabilizer group and the gauge group of the subsystem code, i.e. $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{G}$. We can then map $\mathcal{S}'$ invoking Theorem 1, which shows that the stabilizer generators of $\mathcal{S}$ are locally equivalent to a subset of the stabilizer generators of several copies of KTCs. The simplest way to understand this result is to work it in reverse. Imagine starting with two copies of KTC. There are 16 topological charges, that are generated by the "elementary particles" e^j and m^j, with $j = 1, 2$ labeling the two KTCs. To obtain a subsystem code, one could choose to encode information only in KTC^1, and not enforce the stabilizers of KTC^2. The charges e^2 and m^2 would therefore be gapless and the associated encoded qubit would carry random information. We say that we have "gauged out" the elementary charges e^2 and m^2, and retained e^1 and m^1 as "proper charges". Note that the proper charges do not topologically interact with the gauge charges, which ensures that the information they encode is protected. In this example, $\mathcal{S}$ would be the stabilizer of KTC^1, $\mathcal{S}'$ would be the stabilizer of the two KTCs, and $\mathcal{G}$ would be all the operators acting on KTC^2 and the stabilizers of KTC^1.

A slightly less trivial example can be constructed by choosing a different set of elementary charges. Consider the four fermions $\xi_1 = m^1 \times f^2$, $\xi_2 = e^1 \times f^2$, $\xi_3 = f^1 \times m^2$, and $\xi_4 = f^1 \times e^2$ that generate all the topological charges of the two KTCs. Note that the pairs (ξ_1,ξ_2) and (ξ_3,ξ_4) are canonical in the sense that mutual statistics are semionic in a pair and trivial between particles from distinct pairs. Thus, we can choose to gauge out ξ_3 and ξ_4 and retain ξ_1 and ξ_2 as proper charges to encode information. It is interesting to note that the proper charges here form a chiral anyon model. Our main result shows that whenever a TSC $\mathcal{S}'$ exists with $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{G}$, the corresponding TSSC $(\mathcal{S},\mathcal{G})$ can be generated this way, starting with n copies of KTC, choosing a set of elementary charges, k of which braid trivially with the rest and are gauged out. In addition, the resulting code can be modified by a local quantum circuit.

Note that because $\mathcal{S}$ is a strict subset of $\mathcal{S}'$, this mapping does not take the system to the ground state of the resulting KTCs; in a code state of the TSSC, all elements of $\mathcal{S}$ take value $+1$ but the stabilizers added to $\mathcal{S}$ to arrive at $\mathcal{S}'$ can take any value, i.e. the system is in general not in a $+1$ eigenstate of the added stabilizer generators. These added generators can be measured, which will result in random excitations with trivial proper charge but arbitrary gauge charge. Thus, if one is interested to physically map a TSSC to KTCs, an additional step is required. These excitations can be eliminated by local transformations; simply pairing up elementary excitations in an arbitrary way and fusing each pair into the vacuum. Moreover, because these excitations correspond to gauge charges, this local transformation does not change the encoded information; the different ways of pairing the elementary excitations will only affect the gauge sector of the Hilbert space.

A. Examples

Let us illustrate this strategy with an important family of 2D subsystem codes [25] called topological subsystem color codes (TSCCs). Given the lattice of a TCC, we can inflate each vertex into a triangle as in Fig. 13 a). Qubits are located on the vertices of this inflated lattice, and there is one gauge group generator associated to each pair of sites i, j connected by an edge

$$G_{ij} = \sigma^i \sigma^j \tag{8}$$

with $\sigma = \sigma_x, \sigma_y$, or σ_z for a dashed, dotted, or solid edge respectively. This code admits a set of local stabilizer generators, some of which involve a relatively large number of qubits (up to 24). Excitations are described by two elementary topological charges (ξ_1,ξ_2), both fermions and with semionic mutual statistics, making it a chiral anyon model. The fusion rules are $\xi \times \xi \to 0$ and $\xi_1 \times \xi_2$ is a composite fermion. These topological properties are identical to those of the proper charges ξ_1 and ξ_2 constructed in the previous paragraph. This suggests that we should be able to find a TSC $\mathcal{S}'$ with $\mathcal{S} \subset \mathcal{S}' \subset \mathcal{G}$ and $\mathcal{S}'$ locally equivalent to n copies of KTC with $n \geq 2$. We will present two different ways of obtaining $\mathcal{S}'$ that are geometry independent (i.e., not restricted to the square-octagon lattice).

In the first construction, $\mathcal{S}'$ is the stabilizer of three TCCs on the corresponding non-inflated lattice. Indeed, all we need to do is to rearrange the qubits. The three qubits located at the vertices of each triangle inherit the color label of the neighboring plaquette. We construct

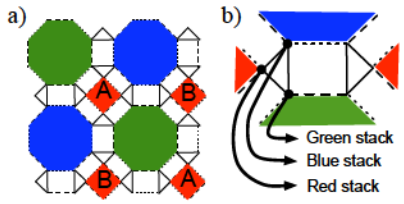


FIG. 13: a) Expanded square-octagon lattice for TSCC. Starting with the lattice of a TCC (see Fig. 11), each vertex is expanded into a triangle. There is one gauge operator Eq. (8) per edge. b) Zoom of a region of the extended lattice and rearrangement of the qubit into three stacks.

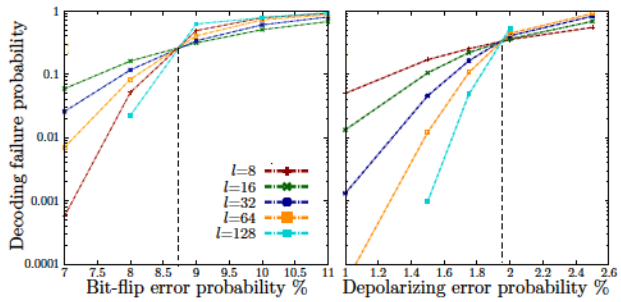


FIG. 14: Decoding failure probability as a function of the error probability of each qubit for the square-octagon TCC (left) and TSCC (right), based on the algorithm of [23]. The different curves illustrate lattices of different linear size l: below a threshold probability (dotted lines), the decoding failure probability decreases with the lattice size, leading to a perfect recovery in the thermodynamic limit.

a stack of three TCC lattices—one per color—each one containing the qubits of that color, see Fig. 13 b). It can be easily verified that this maps the generators of $\mathcal{S}$ to a subset of the generators of the 3 TCCs. We obtain $\mathcal{S}'$ by including the other stabilizer generators of these TCCs. In the second construction, we consider the stabilizer group $\mathcal{S}_z$ generated by the gauge operators of the form $\sigma_z\sigma_z$ (solid edges), which clearly is a subgroup of $\mathcal{Z}(\mathcal{S})$. It follows from the results in [34] that $\mathcal{S}' := \mathcal{S}\mathcal{S}_z$ is a TSC with the same topological charges as a TCC. These two constructions illustrate that the quantum dimension of the intermediate code $\mathcal{S}'$ is not uniquely determined, since in the first case we have $D = 2^6$ and in the second $D = 2^2$.

V. DECODING

When the system is prepared in the ground state of the Hamiltonian Eq. (1), all stabilizers have value $+1$. But in the presence of errors, this will not be the case in general. The problem of decoding a quantum code consists in identifying the most likely recovery to restore the encoded state from partial information coming from the measurement of the stabilizer operators, whose ± 1 outcomes are called error syndrome. Not all codes can be decoded efficiently, but fast approximate algorithms have been devised for KTC. The one presented in [12] uses Edmonds' minimum matching algorithm [35] to find the shortest path that recombines all electric particles in pairs and independently all magnetic particles in pairs. For N qubits, it runs in time N^3. The algorithm proposed in [23] uses renormalization group approximations to find the homological class of errors with the highest probability. It runs in time $\log N$. An efficient decoder was also devised for a topological subsystem color code (TSSC) on a particular lattice [24], but in general each new code requires a tailored decoding technique.

Our techniques can be used to decode any 2D TSCs. Indeed, our main Theorem shows the existence of a transformation that maps an error syndrome of a TSC to a syndrome on the KTCs. Given that syndrome, we can run any of the known decoding algorithms [12, 23] on each of the KTCs, and translate the proposed recovery back to the original TSC. In fact, the idea extends to any TSSCs. As discussed in the proof of our main Theorem, we can always find a set of canonical elementary charges e^i, m^i (these could be fermions for TSSCs) that generate all the topological charges in the code. Then, the decoding problem boils down to matching all elementary defects in pairs, just like for KTC. The advantage here is that decoding does not require an explicit unitary mapping between the codes, but only a mapping between excitations. We have used this technique, combined to the decoding algorithm of [23], for the TCC on the square-octagon lattice of Fig. 11 on a bit-flip channel and found an error threshold of roughly 8.7% (see Fig.14), in good agreement with the Monte Carlo estimate of 10.9% [36] for ideal error correction. We have also used this technique for the TSCC on the square-octagon lattice of Fig. 11 on a depolarizing channel and found an error threshold of roughly 1.95% (see Fig.14), in good agreement with the estimate of 2% [24] for a closely related code in a five-square lattice.

VI. CONCLUSION

We have demonstrated that 2D topological stabilizer codes all belong to one universal topological phase by constructing an explicit local mapping onto multiple copies of Kitaev's topological code. This result also carries to a certain class of 2D subsystem codes, and in particular to all topological subsystem color codes. These local maps enable us to extend many properties of Kitaev's code to all 2D codes, and in particular directly yield efficient decoding algorithms for error correction. It could also have important implications for fault-tolerant quantum computation.

Acknowledgements— We thank Sergey Bravyi for stimulating discussions. This work was partially funded by IARPA QCS program, NSERC, FQRNT, Mprime, Industry Canada, Ontario MRI, MICINN, CAM, and

UCM-BS. Computational resources were provided by RQCHP.

[1] S. Bravyi, M. B. Hastings, and S. Michalakis, J. Math. Phys. **51**, 093512 (2010).

[2] A. Y. Kitaev, Ann. Phys. **303**, 2 (2003).

[3] S. Bravyi and M. B. Hastings. A short proof of stability of topological order under local perturbations. 01 2010.

[4] S. Dusuel, M. Kamfor, R. Orús, K. P. Schmidt, and J. Vidal. Robustness of a perturbed topological phase. *Phys. Rev. Lett.*, 106:107203, Mar 2011.

[5] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, Rev. Mod. Phys. **80**, 1083 (2008).

[6] X. Chen, Z.-C. Gu, and X.-G. Wen, Phys. Rev. B **82** (2010).

[7] M. Hastings and X. Wen, Physical Review B **72** (2005).

[8] E. Knill and R. Laflamme, Phys. Rev. A **55**, 900 (1997).

[9] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, IEEE Trans. Info. Theor. **44**, 1369 (1998).

[10] A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).

[11] M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).

[12] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).

[13] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).

[14] D. Bacon, Phys. Rev. A **73**, 012340 (pages 13) (2006).

[15] H. Bombin and M. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).

[16] Z. Nussinov and G. Ortiz, Phys. Rev. B **77**, 064302 (2008).

[17] R. Alicki, M. Fannes, and M. Horodecki, J. of Phys. A **42**, 065303 (2009).

[18] C. Castelnovo and C. Chamon. Entanglement and topological entropy of the toric code at finite temperature. *Phys. Rev. B*, 76:184442, Nov 2007.

[19] S. Iblisdir, D. Pérez-García, M. Aguado, and J. Pachos. Scaling law for topologically ordered systems at finite temperature. *Phys. Rev. B*, 79:134303, Apr 2009.

[20] S. Bravyi and B. Terhal, New J. Phys. **11**, 043029 (2009).

[21] A. Kay and R. Colbeck, (2008), arXiv:0810.3557.

[22] M. Aguado and G. Vidal, Phys. Rev. Lett. **10**, 070404 (2008).

[23] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).

[24] M. Suchara, S. Bravyi, and B. Terhal, (2010), arXiv:1012.0425.

[25] H. Bombin, Phys. Rev. A **81**, 032301 (2010).

[26] P. Arrighi, V. Nesme, and R. Werner, J. Comput. Syst. Sci. **77**, 372 (2011).

[27] A. Kitaev, Ann. of Phys. **321**, 2 (2006).

[28] J. Eisert and M. Cramer. Single-copy entanglement in critical quantum spin chains. *Phys. Rev. A*, 042112 (2005).

[29] X.-G. Wen, Phys. Rev. Lett. **90**, 16803 (2003).

[30] S. Bravyi, Private communications (2011).

[31] H. Bombin, Phys. Rev. Lett. **105**, 030403 (2010).

[32] H. Bombin (2011), arXiv:1107.2707.

[33] M. Levin and X.-G. Wen Phys. Rev. B **67**, 245316 (2003).

[34] H. Bombin, M. Kargarian, and M. A. Martin-Delgado, Phys. Rev. B **80**, 075111 (2009).

[35] J. Edmonds, Canad. J. Math. **17** (1965).

[36] H. G. Katzgraber, H. Bombin, R. S. Andrist, and M. A. Martin-Delgado, Phys. Rev. A **81**, 012319 (2010).

[37] In Abelian models, the chiral central charge c_- is related to the topological spins θ_i (=1 for bosons, -1 for fermions) through $\sum_i \theta_i = \kappa e^{\pi i c_-/4}$ [27].

Chapitre 6

Article : Kitaev's Z_d-code threshold estimates

Guillaume Duclos-Cianci, David Poulin, *Kitaev's Z_d-code threshold estimates*, Phys. Rev. A, 87, 062338 (2013).

6.1 Contexte

Le code topologique de Kitaev le plus souvent étudié est construit à l'aide de qubits. Or, la construction originale s'applique à n'importe quel groupe fini [28]. Dans le cas des qubits, le groupe de charge observé est $\mathbb{Z}_2$, car chaque particule est sa propre antiparticule. Pour un groupe G quelconque, on peut construire un code de Kitaev à l'aide de qu*d*its [1] où d est l'ordre de G. Une extension simple consiste à prendre les groupes cycliques $\mathbb{Z}_d$, d'ordre d. Dans ce cas, les particules ne sont plus leur propre antiparticule, mais elles sont toujours des anyons abéliens dont la charge est un élément de $\mathbb{Z}_d$. Par conséquent, le problème du décodage ne se réduit plus à celui d'un appariement et la plupart des méthodes usuelles ne s'appliquent plus. Heureusement, la méthode de décodage par renormalisation ne dépend pas de cette structure. En adaptant l'algorithme aux distributions de probabilité sur des *d*its[1], on peut toujours décoder ces codes. Avec cette adaptation, j'ai estimé les seuils de ces différents codes. Comme attendu, pour un taux de bruit donné, le seuil augmente avec d. En effet, plus d est grand, plus une même densité de défauts contient d'information. Par contre, nous ne nous attendions pas à constater que la progression de la valeur des seuils en fonction

1. Lorsque l'on considère des systèmes à d niveaux plutôt que deux (où d est un entier supérieur à deux), on parle de *d*its dans le cas classique et de qu*d*its dans le cas quantique.

de d colle étonnement bien à la borne de hachage. Nous savons que cette borne s'applique aux codes dits « non-dégénérés ». Or, les codes de Kitaev sont hautement dégénérés. Nous n'avons pas d'explication définitive de cet état de fait. Toutefois, cette correspondance porte également le nom de conjecture de Nishimori et a été observée de manière non-rigoureuse en utilisant la méthode des répliques [29]. Il est aussi important de souligner que j'ai été le principal auteur de cette publication, c'est-à-dire que j'ai rédigé la majorité de ce qui se retrouve dans la publication finale. Ces travaux ont été accepté à la conférence *Theory of Quantum Computation, Communication and Cryptography (TQC) 2013*, mais je n'ai pu me présenter à la conférence pour des raisons de santé.

6.2 Résumé

La section I introduit l'article. La section II définit le qudit ainsi que le groupe de Pauli généralisé [30]. Nous révisons certains détails importants dans la section 6.3 ci-bas. La section II introduit aussi le code de Kitaev généralisé. Davantage de détails se trouvent dans la section 6.4. La section III discute des erreurs et du problème du décodage dans ce contexte. Un point important est qu'il ne s'agit plus d'un problème d'appariement. Quelques exemples utiles se trouvent à la section 6.5. La section IV présente l'algorithme en détails, avec la base d'opérateurs pertinente et les règles d'échange de messages. La section V présente les seuils estimés et fait état d'un accord inattendu des résultats avec la borne de hachage. La borne de hachage est présentée à la section 6.6.

6.3 Qudit et groupe de Pauli généralisé

Un qudit est simplement un système quantique à d niveaux, c.-à-d. un système dont l'espace de Hilbert est généré par les états $|g\rangle$, où $0 \leq g < d$. De manière analogue au cas du qubit, nous définissons les opérateurs X et Z tels que X permette de passer d'un état au suivant, parcourant l'espace de Hilbert de manière cyclique, et tels que Z ajoute une phase $\omega^g = e^{i2\pi g/d}$ à l'état $|g\rangle$, cf. Eq. (6.1).

$$X = \sum_{g=0}^{d-1} |g \oplus 1\rangle\langle g|, \qquad Z = \sum_{g=0}^{d-1} \omega^g |g\rangle\langle g|, \tag{6.1}$$

où $\oplus$ désigne l'addition modulo d. L'Eq. (6.2) explicite un exemple pour $d = 4$.

$$X = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad Z = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & i & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -i \end{pmatrix} \tag{6.2}$$

L'équation Eq. (6.3) explicite la relation de commutation de X et Z pour d quelconque.

$$\begin{aligned} ZX\,|g\rangle &= Z\,|g \oplus 1\rangle \\ &= \omega^{g+1}\,|g \oplus 1\rangle \\ &= \omega^{g+1} X\,|g\rangle \\ &= \omega XZ\,|g\rangle \end{aligned} \tag{6.3}$$

Remarquons que ces opérateurs sont unitaires, mais pas hermitiens. Ils ne sont donc pas leur propre inverse, contrairement aux matrices de Pauli sur des qubits.

6.4 Code topologique de Kitaev Z_d

Le code topologique de Kitaev sur des qudits ($\mathbb{Z}_d$-CTK) est défini à l'aide des opérateurs de Pauli généralisés. Considérons un réseau carré où un qudit est placé sur chaque arête. Comme dans le cas du code avec les qubits, nous voulons définir un ensemble de générateurs invariants sous translation qui commutent deux à deux. Or, les opérateurs de Pauli généralisés ne commutent ni n'anti-commutent en général (cf. Eq. (6.3)). Bêtement calquer les opérateurs plaquettes et sites du CTK sur des qubits ne fonctionne donc pas. En effet, un opérateur plaquette et un opérateur site partageant deux qudits ne commuteront pas. Pour remédier à ce problème, nous exploitons le fait que ces opérateurs ne sont pas leur propre inverse, contrairement au cas des qubits. La figure 1 de l'article propose une définition pour les opérateurs plaquettes et sites. Cette solution n'est pas unique : toute rotation ou réflexion de ces opérateurs fonctionne aussi bien. Deux critères doivent être respectés. D'une part, chaque opérateur plaquette (site) doit avoir deux Z (X) et deux $Z^\dagger$ ($X^\dagger$) comme facteurs et ceux-ci doivent être disposés tel que l'opérateur de gauche soit l'inverse de celui de droite et que l'opérateur du haut soit l'inverse de celui du bas. Ceci assure que les produits de générateurs de chaque type donnent bien des boucles du réseau direct ou dual. D'autre part, les opérateurs doivent être disposés de sorte que les générateurs plaquettes et sites commutent deux à deux. L'intérêt de cette définition est que le code $\mathbb{Z}_2$-CTK redonne bien le code de Kitaev avec des qubits.

Tel que défini ci-haut, le code ne peut être écrit comme le fondamental d'un hamiltonien, car les générateurs ne sont pas des opérateurs hermitiens. Pour ce faire, il suffit de les combiner de manière à ce que leurs nouvelles valeurs propres soient réelles :

$$H = -\frac{1}{2}\sum_s (A_s + A_s^\dagger) - \frac{1}{2i}\sum_s (A_s - A_s^\dagger) - \frac{1}{2}\sum_p (B_p + B_p^\dagger) - \frac{1}{2i}\sum_p (B_p - B_p^\dagger), \tag{6.4}$$

où A_s désigne les opérateurs sites et B_p, les opérateurs plaquettes. Comme les valeurs propres des générateurs sont de la forme ω^g, il s'ensuit que les nouveaux opérateurs ont des valeurs propres de la forme $\cos(2\pi g/d)$ ou bien $\sin(2\pi g/d)$, avec $0 \leq g < d$. Pour simplifier les discussions, nous désignons les valeurs propres par « g » tout simplement.

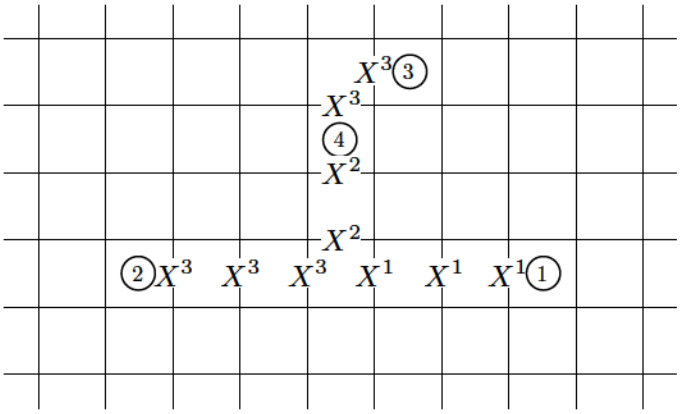


FIGURE 6.1 Amas de défauts créé par une erreur sur un $\mathbb{Z}_5$-CTK. Les défauts n'apparaissent pas qu'aux extrémités. De plus, l'erreur peut se ramifier sans laisser de traces.

6.5 Erreurs, charge topologique et décodage

Dans le cas des $\mathbb{Z}_d$-CTK, nous étudions les erreurs provenant du groupe de Pauli généralisé. La figure 2 de l'article montre comment une erreur de type X crée une paire particule - antiparticule. Le groupe de charge est cyclique d'ordre d et avec notre choix d'étiqueter les valeurs propres par le nombre g, il est représenté par $\mathbb{Z}_d$. Une différence notoire par rapport au $\mathbb{Z}_2$-CTK est qu'une erreur agissant sur un chemin n'aura en général pas que des excitations à ses extrémités, comme le montre la Fig. 6.1. De plus, pour compliquer davantage la situation, une erreur sur un chemin peut se ramifier sans laisser de traces apparentes. Pour cette raison le décodage ne consiste plus en un problème d'appariement deux à deux. Il faut plutôt regrouper les particules en amas distincts. Les techniques usuelles de décodage pour le $\mathbb{Z}_2$-CTK exploitaient précisément cette propriété de défauts en paires. Or, le décodeur RG n'a pas ce problème, celui-ci se basant plutôt sur une approximation des flots de charge à travers le réseau.

6.6 Borne de hachage

La borne de hachage est une borne supérieure sur les performances des codes classiques. Les codes quantiques non-dégénérés obéissent aussi à une borne similaire. Dans le cas des codes CSS non-dégénérés elle peut même être simplement exprimée en fonction du cas classique. Par contre, il n'y a aucune raison de croire que les codes dégénérés y sont contraints.

La borne se déduit d'un argument de comptage. Si nous supposons un bruit classique d'inversion symétrique avec probabilité p, nous nous attendons à ce qu'une erreur ait en moyenne un poids np (nombre d'inversions). Si les différents mots codes sont isolés les uns des autres par une distance d'au moins $2np$ dans l'espace des mots codes, alors il est possible de décoder correctement avec grande probabilité. Remarquons que le nombre de façons de placer np erreurs sur une chaîne de n bits est $\binom{n}{np} \approx 2^{nH_2(p)}$, où cette dernière approximation est obtenue par la formule de Stirling et où $H_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$ est l'entropie de Shannon binaire. Nous en déduisons que chaque mot code doit occuper une « sphère » (hypercube) de rayon $nH_2(p)$ dans l'espace des vecteurs binaires. Nous avons donc 2^k mots codes occupant chacun un volume $2^{nH_2(p)}$. Le volume total étant 2^n, nous obtenons l'inégalité Eq. (6.5). En prenant $n \to \infty$, tout en posant un ratio de qubits encodés $\frac{k}{n} \to 0$, nous obtenons la borne de hachage classique Eq. (6.6) (en prenant le logarithme) : le taux de bruit à compter duquel il est possible d'encoder de l'information de manière asymptotiquement sûre. Dans le cas des codes CSS non-dégénérés, nous pouvons interpréter le code quantique comme deux codes classiques simultanés, l'un protégeant des erreurs X et l'autre protégeant des erreurs Z, c.-à-d. chaque mot code doit occuper une sphère de rayon $2nH_2(p)$. Nous retrouvons alors l'Eq. (6.7). Dans le cas classique, un rendement non-nul est possible tant que $p < 1/2$, alors que le cas quantique CSS non-dégénéré requiert $p < 0.11\ldots$

$$2^n \geq 2^{n(\frac{k}{n}+H_2(p))} \tag{6.5}$$

$$1 \geq H_2(p), \quad \text{cas classique.} \tag{6.6}$$

$$1 \geq 2H_2(p), \quad \text{cas quantique CSS non-dégénéré.} \tag{6.7}$$

d	Borne approximative
2	0.11
3	0.16
4	0.19
5	0.21
6	0.23

Tableau 6.1 Valeurs de la borne de hachage pour les codes CSS non-dégénérés sur des qudits.

La borne peut être généralisée aux codes classiques utilisant des dits et quantiques utilisant des qudits. Pour ce faire, il suffit de remplacer l'entropie binaire par une entropie

en base d : $H_d = -\sum p \log_d p$. Les dérivations précédentes se généralisent directement. Les valeurs de la borne pour $2 \leq d \leq 6$ sont énumérées au Tab. 6.1.

6.7 Erratum

Dans les sections II et III de l'article, nous « symétrisons » les générateurs du stabilisateur, car ils ne sont pas hermitiens. Toutefois, le changement proposé est incomplet, il faut lui ajouter les combinaisons impaires, $\frac{1}{2i}(s - s^\dagger)$, de telle sorte que le nombre total de générateurs soit préservé. De plus, il est faux de prétendre que les valeurs propres ω^a et $\cos(2\pi a/d)$ sont équivalentes, car la fonction cos est paire. Pour vraiment avoir une équivalence, il faut obtenir la paire de valeurs $\cos(2\pi a/d)$ et $\sin(2\pi a/d)$.

La formule ajustée aux données (*fit*) de la figure 4 de l'article est en fait

$$p_{dec} = a + b(p_{phys} - p_{th})L^{1/\nu}. \tag{6.8}$$

6.8 Article

Kitaev's $\mathbb{Z}_d$-Codes Threshold Estimates

Guillaume Duclos-Cianci[1] and David Poulin[1]

[1]*Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1, Canada*

(Dated: April 19, 2015)

We study the quantum error correction threshold of Kitaev's
to a generalized bit-flip noise. This problem requires novel decc
we generalize the renormalization group method we previously
codes.

PACS numbers: 03.67.Pp,03.67.-a

I. INTRODUCTION

Kitaev's topological code (KTC) [3] on qubits is the archetypical topological code and has been extensively studied. As explained in Kitaev's original paper [3], this construction applies to any group. Much less is known about these generalizations, and in this paper we investigate the quantum error correction (QEC) thresholds of the KTCs built with the groups $\mathbb{Z}_d$ [4–7], where $d \geq 2$. We label these as $\mathbb{Z}_d$-KTC, so the original code on qubits corresponds to $\mathbb{Z}_2$-KTC. These codes are well suited for arrays of d-level quantum systems, qudits. Qudits occur very naturally in nature: e.g., superconducting qubits [8], Rydberg atoms [9], orbital states of light [10], etc. have multiple levels and their Hilbert space needs to be truncated to obtain qubits.

As explained in [11], $\mathbb{Z}_2$-KTC can be decoded by a binary perfect matching algorithm [12], since every particle is its own anti-particle in this model. Because this is not the case for $d > 2$, other techniques are required and for this purpose we generalize the renormalization group (RG) soft decoder that we introduced in [1, 2]. Our numerical simulations show that the threshold increases monotonically with d and appears to follow the general trend of the qudit hashing bound.

This paper is organized as follows. First, we introduce a generalized Pauli group (see [13, 14] for more details), stabilizer codes, and $\mathbb{Z}_d$-Kitaev's toric code. Next, we briefly review the decoding problem of these systems and show how the RG decoder applies in this case. Finally, we present the numerical results and close with a discussion.

II. $\mathbb{Z}_d$ GENERALIZATION OF KITAEV'S TORIC CODE

In this section, we review the definition of $\mathbb{Z}_d$-KTC and show that many features of KTC on qubits extend to them. Since we will be working with qudits, we introduce a generalized Pauli group. The Hilbert space of a qudit, $\mathcal{H}_d$, is spanned by the states $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\}$. We define the operators X and Z such that

$$\begin{aligned} X|g\rangle &= |g \oplus 1\rangle, \\ Z|g\rangle &= \omega^g |g\rangle, \end{aligned} \tag{1}$$

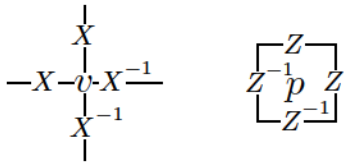


FIG. 1: $\mathbb{Z}_d$-KTC stabilizer generators. To each vertex v, we associate an operator A_v(left) and to each plaquette p, we associate an operator B_p (right).

where $0 \leq g < d$, "$\oplus$" denotes addition modulo d, and $\omega = e^{i2\pi/d}$. The generalized Pauli group is generated by X, Z, and a phase, i.e., $\mathcal{P}_d = \langle \omega, X, Z \rangle$ if d is odd and $\mathcal{P}_d = \langle \omega^{1/2}, X, Z \rangle$ if d is even (XZ has order $2d$ in this case). From the definitions of Eq. (1), we deduce the following properties

$$\begin{aligned} X^a|g\rangle &= |g \oplus a\rangle, \\ Z^a|g\rangle &= \omega^{ag}|g\rangle, \\ ZX|g\rangle &= \omega XZ|g\rangle, \\ Z^a X^b|g\rangle &= \omega^{ab} X^b Z^a |g\rangle. \end{aligned} \tag{2}$$

Lastly, we define the n-qudit Pauli group $\mathcal{P}_d^n \equiv \mathcal{P}_d^{\otimes n}$ as the n-fold tensor product of $\mathcal{P}_d$.

The stabilizer group $\mathcal{S}$ is an ablian subgroup of $\mathcal{P}_d^n$. The code is defined as the simultaneous $+1$ eigenspace of all stabilizers. Note that even though the generalized Pauli operators are unitary, they are not hermitian in general so do not correspond to physical observables. However, the operator $\frac{1}{2}(s + s^\dagger)$ is hermitian and can be measured. Since s has eigenvalues ω^a, $\frac{1}{2}(s+s^\dagger)$ has eigenvalues $\frac{1}{2}(\omega^a + \omega^{-a}) = \cos(2\pi a/d)$ which are in one-to-one correspondence with the eigenvalues of s.

With these definitions in place, we present a generalization of KTC on qudits, which we call $\mathbb{Z}_d$-KTC, using Kitaev's original construction [3] on the cyclic groups $\mathbb{Z}_d$ with $d \geq 2$. The system is a square lattice of linear size L with periodic boundary conditions. Each edge is occupied by a qudit, so there are in total $n = 2L^2$ qudits. We define vertex operators A_v and plaquette operators B_p as shown in Fig. 1. There is one such operator for each vertex and each plaquette. We verify that they commute using the last line of Eq. (2). These operators generate the stabilizer group $\mathcal{S} = \langle A_v, B_p \rangle$ and the code is spanned

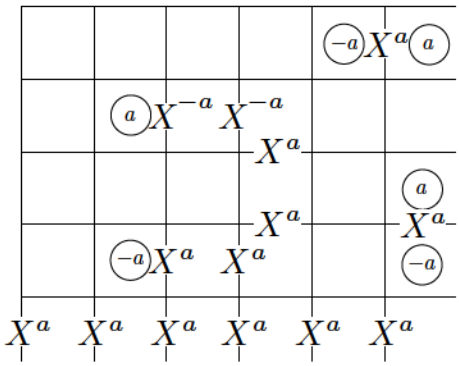


FIG. 2: Plaquette defects created by the application of some power of X. The values a $(-a)$ in the plaquettes are such that the eigenvalue of the corresponding B_p is ω^a (ω^{-a}). By choosing appropriately the powers of X, we can build string operators with defects only on their endpoints. Non-trivial cocyles of X^a correspond to $\overline{X}^a$ logical operators.

by the simultaneous $+1$ eigenstates of the stabilizer generators.

Figure 2 illustrates how applying some power of X on a codestate creates defects on the lattice. Indeed, X^a applied on some qudit does not commute with the two plaquette operators involving that qudit. The eigenvalues of the plaquettes to the north or east of the error will change from 1 to ω^a, and those of the plaquettes to the south or west will change from 1 to ω^{-a}. One can show that the defects thus created are topological charges; we associate the charge a to a plaquette defect corresponding to an eigenvalue ω^a of that plaquette. With this choice of labeling, the charge group restricted to plaquettes is $\mathbb{Z}_d$ with addition.

From these simple facts, it follows that string operators can be built with defects attached only to their endpoints (these strings actually live on the dual lattice, just like in KTC). This requires a careful choice of the powers of X on the qudits along the string such that the total charge in each plaquette is 0 except on its endpoints. For instance, one can adopt the convention that power a is used when heading north or east, and $-a$ when heading south or west. Moreover, we can verify that non-trivial cocycles (loops on the dual lattice, see Fig. 2) of any power of X obeying this convention commute with the stabilizer. These operators are not in the stabilizer as all the vertex generators of Fig. 1 are trivial cocyles. It follows that such operators, e.g. the one found at the bottom of Fig. 2, are logical operators (for any value of a).

A similar analysis holds for defects created by powers of Z operators. In this case, the defects live on vertices and string operators, on the direct lattice. Also, non-trivial cycles of any power of Z are logical operators. From the form of the logical operators, we directly deduce that there are two qudits encoded in the code space. Again, this is analogous to the case of KTC.

III. $\mathbb{Z}_d$-KTC DECODING

We are now interested in the problem of error correcting $\mathbb{Z}_d$-KTCs for $d > 2$. In our study, we consider a simple noise model that generalizes the independent symmetric bit-flip channel to qudits : with probability $1 - p_{\text{phys}}$, the qudit remains unaffected and with probability p_{phys}, we apply at random (uniformely distributed) one of $X, X^2, \ldots, X^{d-1}$ (see Appendix A). Suppose an error $E \in \mathcal{P}_d^n$ occurs on a code state. It creates defects on the lattice and by measuring the eigenvalues of every $\frac{1}{2}(A_v + A_v^\dagger)$ and $\frac{1}{2}(B_p + B_p^\dagger)$ we can learn the position and charge of each defect. The role of the decoder is to bring the system back in the code space by applying a correcting Pauli operator, $C \in \mathcal{P}_d^n$. However, care must be taken in choosing an appropriate correcting operation. Indeed, if the operator CE resulting from the combination of the error and the recovery is an element of $\mathcal{S}$, the state is unaffected. However, if CE is a non-trivial logical operator, then the system is returned to the code space but potentially in a different code state, so the information is corrupted.

Any operator $E \in \mathcal{P}_d^n$ creating the measured configuration of defects is a potential error. However, we classify these operators by their logical effect on the code space: two operators E_1, E_2 with the same configuration of defects are equivalent iff $E_2^\dagger E_1$ has a trivial effect on the code, i.e. $E_1 \sim E_2$ iff $E_2^\dagger E_1 \in \mathcal{S}$. Note that since E_1 and E_2 lead to the same defect configuration, $E_2^\dagger E_1$ creates no defect, or equivalently, E_1 creates some defects that $E_2^\dagger$ annihilates.

Given a measured defect configuration, the decoder seeks for the best correction among the set of all errors which would lead to this defect configuration. One strategy would be to identify the error from this set that has the largest probability $\mathcal{P}(E)$, where the probability of an error is specified by the physical noise model, in our case the symmetric bit-flip channel. This turns out not to be optimal however, because some errors have equivalent effects on all code states. Thus, the decoder should instead seek for the most likely equivalence class of errors. The probability of an equivalence class of errors is obtained by summing over the probability of each error within a class. Given these probabilities, the optimal correction consists in applying the adjoint of any representative of the class with maximal probability.

IV. RG DECODER GENERALIZATION TO $\mathbb{Z}_d$-KTC

Unfortunately, the above procedure cannot be realized efficiently in general since the number of errors in each equivalence class scales exponentially with the system size. In [1, 2], we introduced a renormalization group soft decoder (RG decoder) that efficiently approximates the exact calculation (see [15] for a related scheme). The

FIG. 3: (a) The lattice is cut into unit cells containing ten qudits (edges). The renormalization process takes the defect configuration and the noise model on a unit cell as inputs and outputs a two-qudit distribution (white disks) which corresponds to a probability on the charge flow through the corresponding boundaries. Green disks represent plaquette operators. The plaquette corresponding to the green circle is replaced by the product of all four plaquettes of the unit cell, such that its eigenvalue gives the total charge of the cell. This value is only going to be used in the next round of RG (larger green disk). (b) Labeling convention for qudits in Eq. (3)

general idea is to cut the lattice into small unit cells (e.g. 2×2 sub-lattices) and to "distill" from each cell an effective two-qubit noise model, c.f. Fig. 3(a). This is realized by keeping track of the flow of charges through the cell and summing over the microscopic details leading to this flow. This has the effect of shrinking the lattice linear size by a constant factor (k for cells of size $k \times k$). Recursing on this process, one can shrink the lattice to a constant, manageable, size where the exact decoding can be performed. With appropriate simple modifications, this method can be used for charges over $\mathbb{Z}_d$.

There are two technical difficulties in realizing the above heuristic description, which are both caused by charge conservation. First, because the unit cells share boundaries, the flow of charge through one boundary of a cell should be equal and opposite to the flow of charge of the corresponding boundary of the neighbouring cell. Thus, the variable corresponding to charge flows in each cell are highly constrained. This problem is easily circumvented by keeping only track of the flow of charge through the northern and the western boundary of each cell, i.e. by eliminating this redundancy.

Second, the sum of the charge flow through the boundaries of a cell must be equal to its total charge, revealed by the syndrome measurement. This once again sets a hard constraint between the variables corresponding to the charge flows, which would in principle require a probability distribution that correlates all the variables of the system. This cannot be realized efficiently, so we must resort to some approximation. As a first approximation, we choose to ignore the cross-cell correlations, and keep only marginal probabilities on the flows associated to a given cell (we keep a probability distribution that involves the northern and western boundary only). To diminish the effect of these correlations we are neglecting, we let the charge inside a unit cell fluctuate. For each unit cell, we measure all but one of the plaquettes it encloses. This remaining plaquette thus determines the total charge of the unit cell, and indeed we can substitute the corresponding stabilizer generator by a plaquette enclosing the entire unit cell (obtained by multiplying all the plaquette operators contained in the unit cell). This new stabilizer generator represents a renormalized charge.

This procedure is illustrated on Fig. 3(a) where green disks represent plaquettes that are measured and the green circle represents the plaquette that is left fluctuating. This green circle is replaced by the larger, renormalized green disk (on the right) that is used in the next RG step. The white disks on this figure each represent a probability distribution on charge flow, or equivalently a two-qudit probability distribution. Thus, after one round of RG, we are left with a smaller lattice and both renormalized charges and renormalized noise models.

Equation (3) lists a set of generators for all X operators living on a unit cell (see Fig. 3(b) for labelling). This basis will be used to decompose any X-type error contained on the unit cell. These operators are defined in accordance to the renormalization process itself as we now explain. The T_i operators are used to build a representative error with the appropriate defect configuration. Indeed, only the T_i operators of Eq. (3) do not commute with all three plaquette operators in the unit cell (green disks of Fig. 3(a)). Label the defect configuration on a unit cell as $\vec{a} = (a_0, a_1, a_2)$, where a_0 is the charge of the north-west plaquette, a_1 is the charge of the north-east one, and a_2 is the charge of the south-west one. Then, the Pauli operator $t(\vec{a}) = T_0^{a_0} T_1^{a_1} T_2^{a_2}$ creates the defect configuration $\vec{a}$. Moreover, given a defect configuration $\vec{a}$, every potential error has to contain this product in its decomposition on basis Eq. (3) since only the T_i operators do not commute with plaquettes. The L_i operators characterize the flow of charge through the northen and western boundaries, so the two-qudit output distribution of a RG round is precisely the probability distribution over these two operators. The S_i operators are stabilizer operators (or parts of stabilizer generators supported on the unit cell). They only deform strings without changing their defect configuration or their associated charge flow. Lastly, the E_i operators correspond to charge flowing through the southern and eastern boundaries into the plaquette operator that is left out. Thus, they are responsible for the charge fluctuation inside the unit cell and they are summed over.

$$\begin{aligned} S_0 &= X_0 X_2^{-1} X_3^{-1} & T_0 &= X_4 X_7^{-1} \\ S_1 &= X_1 X_4^{-1} X_5^{-1} & T_1 &= X_6 \\ S_2 &= X_3 X_4 X_6^{-1} X_7^{-1} & T_2 &= X_7^{-1} \\ & & & \\ E_0 &= X_6 X_8 & L_0 &= X_2 X_6 \\ E_1 &= X_7^{-1} X_9^{-1} & L_1 &= X_5 X_7 \end{aligned} \tag{3}$$

With these definitions, we can formally describe a RG round that starts with a defect configuration $\vec{a}$, and com-

putes the marginal probability of each $l \in \langle L_0, L_1\rangle$ conditioned on the measured defect configuration,

$$\mathcal{P}(l) = \sum_{e\in\langle E_0,E_1\rangle} \sum_{s\in\langle S_0,S_1,S_2\rangle} \mathcal{P}(tles), \qquad (4)$$

where $t = T^{a_0}T^{a_1}T^{a_2}$ is given by the defect configuration and $\mathcal{P}(tles)$ is the probability assigned to the error $E = tles$ by the noise model. The complexity of decoding a unit cell is given by the number of operators that are considered in Eq. (4): $|\langle L_0, L_1\rangle|\cdot|\langle E_0, E_1\rangle|\cdot|\langle S_0, S_1, S_2\rangle|$. Since all L_i, E_i and S_i have order d, the complexity is the constant d^7. For different unit cell sizes, the complexity is still a power of d, but with a different exponent which depends on the number of qudits in the cell and the number of measured stabilizer generators. Moreover, the number of unit cells to decode in a given round of RG is given by $(L/k)^2$ where k and L are the linear sizes of the unit cell and the global lattice, respectively. Thus, the complexity of a step of RG goes as $d^c(L/k)^2$ for some constants c and k that depend on the choice of unit cell. Of course, the RG calculations on different cells can be executed in parallel.

The procedure we have described above to evade the correlations caused by local charge conservation is only a heuristic, and can be improved using belief propagation (BP). Roughly, the role of BP is to ensure consistency between the marginal probability of qubits located at the boundary of two or more unit cells, e.g. qudits 0, 1, 8 and 9 (see Fig. 3(b) for labeling). First, given a defect configuration inside a unit cell, one can compute the marginal error probability $\mathcal{P}_q(tles|_q)$ for each qudit q, obtained by taking a marginal of $\mathcal{P}(tles)$. These are called messages and denoted $m_q^{\text{out}}(p)$, where q labels a qudit and p is a one-qudit Pauli operator. These outgoing messages are then exchanged between neighbouring cells, and become incoming messages, e.g. a cell c sends to its northern neighbour c' the message m_0^{out} that becomes m_9^{in} in c', and receives from c' the message m_9^{out} that becomes m_0^{in} in c. Subsequent rounds of messages can be calculated using the received messages, following the prescription

$$m_q^{\text{out}}(p) \leftarrow \sum_{l,s,e} \delta(tles|_q, p)\frac{\mathcal{P}(tles)}{\mathcal{P}_q(tles|_q)} \prod_{q'\neq q} m_{q'}^{\text{in}}(tles|_{q'}), \qquad (5)$$

Here, $q, q' \in \{0, 1, 8, 9\}$, $tles|_q$ is the restriction to qudit q of the Pauli operator $tles$ and $\mathcal{P}_q$ is the marginal on qudit q of the noise model as above. BP can be iterated a few times (e.g. three rounds) before executing a RG step. This has the effect of replacing Eq. (4) by

$$\mathcal{P}(l) = \sum_{e\in\langle E_0,E_1\rangle} \sum_{s\in\langle S_0S_1S_2\rangle} \mathcal{P}(tles)\prod_q m_q^{\text{in}}(tles|_q). \qquad (6)$$

V. NUMERICAL RESULTS

In this section, we present our numerical estimates of the thresholds of $\mathbb{Z}_d$-KTCs for $2 \le d \le 6$ subject to the generalized bit-flip noise model introduced in the previous section. The threshold is defined as the value of the physical noise rate p_{phys} below which the decoding error probability p_{dec} can be made arbitrarily small by increasing the lattice size L.

The simulations were performed as follows. For various values of d, L and p_{phys}, specifiying a $\mathbb{Z}_d$-KTC of linear size L subject to a noise of parameter p_{phys}, we performed a Monte Carlo simulation to estimate the decoding error probability p_{dec}. We used sample sizes of the order of 10^4. For a fixed value of d, we plotted estimates of p_{dec} vs p_{phys} for different values of L. We then used the fitting model

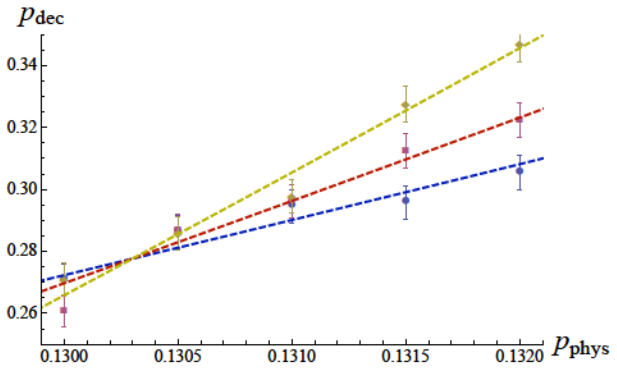


FIG. 4: (Color online) Threshold estimation for $\mathbb{Z}_3$-KTC. The x-axis represents physical error rate and the y-axis, decoding error rate. The blue dots, red squares and yellow diamonds correspond to $L = 32$, $L = 64$ and $L = 128$ respectively. The fitting curve used is $p_{dec} = (p_{phys} - p_{th})L^{1/\nu}$. In this case, we find $p_{th} = 0.13(0)$.

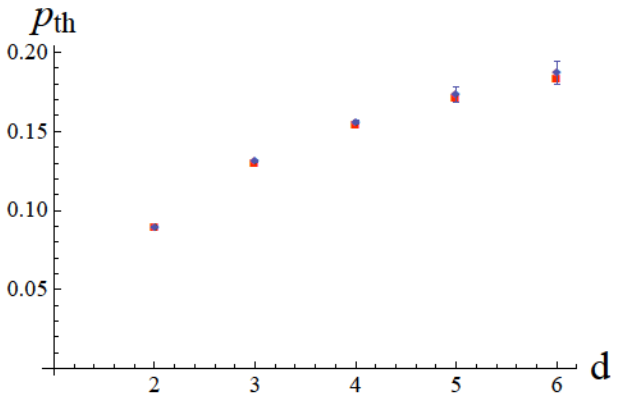


FIG. 5: (Color online) The blue diamonds are the values extracted by fitting the threshold values for $2 \le n \le 6$ (see Fig. 4 for example). The red squares are obtained via the generalized hashing bound (see text) rescaled by a common factor $\alpha = p_{th}(2)/C_2 \approx 0.81$. The error bars are (pessimistically) obtained e.g. by replacing each line in Fig. 4 by a stripe of width equal to the statistical error bars, and determining the values of p_{phys} above and below the crossing point where the strips cease to overlap. We do not report the fitting parameter ν because they are too sensitive to statistical fluctuations and therefore unreliable in our study.

$p_{\text{dec}} = (p_{\text{phys}} - p_{\text{th}})L^{1/\nu}$ (see [11, 16] for more details) to estimate the value of the threshold. As an example, we plotted the results and the fits for $\mathbb{Z}_3$-KTC on Fig. 4.

Repeating this for $3 \leq d \leq 6$ (2 was studied in [1, 2]), Fig. 5 shows p_{th} as a function of d. Heuristically, we did expect that the value of p_{th} increases with d. Indeed, if we imagine simulating a qudit using $\log_2 d$ qubits, a fixed noise rate for increasing values of d translates into a decreased noise rate per qubit. Moreover, it was reported in [17] that the performance of BP for $\mathbb{Z}_d$-KTC, which is very poor in the qubit case, is greatly increased as d grows.

It is intringuing to note that for $\mathbb{Z}_2$-KTC subject to bit-flip or depolarizing noise, p_{th} is numerically very close to the hashing bound [11, 16, 18] (and so do other topological codes [19]). The hashing bound, obtained by a simple packing argument [20], states that for non-degenerate CSS codes,

$$0 \leq 1 - 2H_2(p), \tag{7}$$

where H_2 is the binary entropy: $H_2(p) = -(1-p)\log_2(1-p) - \log_2 p$. From Eq. (7), one can calculate the saturating point $C_2 \approx 0.110$ which is indeed quite close to the optimal threshold of the $\mathbb{Z}_2$-KTC subject to independent bit-flip and phase-flip errors, $p_{th}(2) \approx 0.109(4)$ [11, 16]. This near coincidence is intriguing given that topological codes are highly degenerate, so there is no reason they should obey the hashing bound. Of course, the decoder we are using here is sub-optimal, so the threshold we find $p_{th}(2) \approx 0.89(6)$ is a smaller fraction $\alpha = p_{\text{th}}(2)/C_2 \approx 0.81(4)$ of the hashing bound.

For qudits, the hashing bound is

$$0 \leq 1 - 2H_d(p), \tag{8}$$
$$\text{with} \quad H_d(p) = -(1-p)\log_d(1-p) - p\log_d\frac{p}{d-1}.$$

In this case, we find $C_3 \approx 0.159, C_4 \approx 0.189$ and so on. Figure 5 shows the threshold $p_{\text{th}}(d)$ obtained with the RG decoder as well as a rescaled hashing bound αC_d where α is determined by the $\mathbb{Z}_2$ fit. The agreement is both unexplained and surprisingly good. Note also that even though our decoder is sub-optimal, $p_{th}(d+1) > C_d$ for all d we have studied, which strongly support the claim that the threshold increases with d.

VI. CONCLUSION

In this paper, we presented a generalization of the renormalization group decoder of [1, 2] to Kitaev topological codes built with the groups $\mathbb{Z}_d$. Our numerical results show that the threshold value increases as a function of the local dimension d. Moreover, its behaviour is in very good agreement with a scaling predicted by the hashing bound. This trend could be confirmed by more accurate numerical estimates using a mapping to a statistical mechanics model, which does not require solving the decoding problem [11, 18]. A theoretical understanding of this behavior is also desirable. Lastly, estimating the threshold in the presence of measurement error and detailed syndrome measurement circuits on qudits remains an interesting open question.

VII. ACKNOWLEDGEMENTS

We would like to thank Jonas Anderson for useful discussions regarding the generalized hashing bound. We also thank Simon Burton, Courtney Brell and Stephen Bartlett for enlightening discussions of Kitaev's construction [3]. Computational resources were provided by Calcul Québec and Compute Canada. This work was partially funded by NSERC and by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract D11PC20167. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

[1] G. Duclos-Cianci and D. Poulin, Physical Review Letters **104**, 050504 (2009).
[2] G. Duclos-Cianci and D. Poulin, Information Theory Workshop (ITW) p. 1 (2010).
[3] A. Kitaev, annals phys. **303**, 2 (2003).
[4] S. S. Bullock and G. K. Brennen, Journal of Physics A Mathematical General **40**, 3481 (2007).
[5] H. Bombin and M. A. Martin-Delgado, Journal of Mathematical Physics **48**, 052105 (2007).
[6] M. D. Schulz, S. Dusuel, R. Orús, J. Vidal, and K. P. Schmidt, New Journal of Physics **14**, 025005 (2012).
[7] O. Viyuela, A. Rivas, and M. A. Martin-Delgado, New Journal of Physics **14**, 033044 (2012).
[8] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, eprint arXiv:cond-mat/0703002 (2007).
[9] A. Nussenzweig, J. Hare, A. M. Steinberg, L. Moi, M. Gross, and S. Haroche, EPL (Europhysics Letters) **14**, 755 (1991).
[10] A. Vaziri, J.-W. Pan, T. Jennewein, G. Weihs, and A. Zeilinger, Phys. Rev. Lett. **91**, 227902 (2003).
[11] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452 (2002).
[12] J. Edmonds, Canadian Journal of Mathematics **17**, 449 (1965).

[13] E. Knill, eprint arXiv (1996), quant-ph/9608048.
[14] D. Gottesman, Chaos, Solitons, and Fractals **10**, 1749 (1999).
[15] S. Bravyi and J. Haah, ArXiv e-prints (2011), 1112.3252.
[16] J. W. Harrington, Ph.D. thesis, California Insitute of Technology (2004).
[17] I. Andriyanova, D. Maurice, and J.-P. Tillich, ArXiv e-prints (2012), 1202.3338.
[18] H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, Physical Review X **2**, 021004 (2012).
[19] H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, Phys. Rev. Lett. **103**, 090501 (2009).
[20] A. Ekert and C. Macchiavello, Phys. Rev. Lett. **77**, 2585 (1996).

Appendix A: Generalized bit-flip channel

The noise model described in section III can be seen as emerging from a qudit depolarizing channel. To show this, consider the depolarizing channel which as the same form for qudits as for qubits:

$$\begin{aligned}\epsilon_d(\rho) &= (1-p)\rho + p\frac{I}{d}, \\ &= (1-p)\rho + \frac{p}{d^2}\sum_{i,j=0}^{d-1} X^i Z^j \rho Z^{-j} X^{-i}. \end{aligned} \tag{A1}$$

From this last expression, we see that the probability that error X^i and Z^j occur simultaneously is $\mathcal{P}(X^i, Z^j) = \frac{p}{d^2}$, except when $i = j = 0$ when $\mathcal{P}(I, I) = (1 - p + \frac{p}{d})$. Because we use a CSS code, we can correct X-type and Z-type errors independently. The code correcting X-type errors sees a noise model that is the marginal of $\mathcal{P}(X^i, Z^j)$, i.e. $\mathcal{P}(X^i) = \sum_j \mathcal{P}(X^i, Z^j)$, resulting in the channel

$$\epsilon_x(\rho) = (1 - p + \frac{p}{d})\rho + \frac{p}{d}\sum_{i,j=1}^{d-1} X^i \rho X^{-i} \tag{A2}$$

and similarly for Z.

The generalized bit-flip channel introduced in Section III has the form

$$\epsilon_{bf}(\rho) = (1-p)\rho + \frac{p}{d-1}\sum_{i,j=0}^{d-1} X^i \rho X^{-i}. \tag{A3}$$

Comparing Eq. (A2) and Eq. (A3), we see that performing the following subsitution in ϵ_{bf} yields ϵ_x:

$$p \to (1 - d^{-1})p$$

This type of noise model is typical in information theory to assess the robustness of different codes and architectures. While the depolarizing noise model Eq. (A1) is well motivated physically by its symmetry—i.e. it is the least biased noise model— its marginal, the generalized bit-flip channel Eq. (A3) is somewhat more artificial. However, a coding scheme capable of correcting a generalize bit-flip channel of strength-p for both X- and Z-type errors will also error correct a depolarizing channel of strength $p' = 3p/2$. Moreover, a coding scheme that exploits the correlations between X- and Z-type errors may even tolerate a depolarizing rate larger than this. For more discussions on physically justified noise model, see [7].

Chapitre 7

Article : Fault-tolerant renormalization group decoder for Abelian topological codes

Guillaume Duclos-Cianci, David Poulin, *Fault-Tolerant Renormalization Group Decoder for Abelian Topological Codes*, Quant. Inf. Comp. Vol 14, No 9&10, pp0721-0740 (2014).

7.1 Contexte

Durant ma maîtrise, j'ai développé et programmé un algorithme de décodage pour le code de Kitaev basé sur des idées de renormalisation. Cet algorithme a été conçu dans le cadre de la correction d'erreurs quantique où les mesures de syndrome sont supposées parfaites. Dans ce cas, il se formule en deux dimensions. Par contre, en pratique, la mesure des syndromes est toujours elle-même sujette à des erreurs. Lorsque cette source d'erreurs additionnelle est prise en compte, on parle plutôt de décodage tolérant aux fautes. Il a été montré que le décodage tolérant aux fautes peut être formulé comme un problème qualitativement similaire sur un réseau 3D [4], c.-à-d. un problème d'appariement des défauts. Or, les principes derrière l'algorithme basé sur la renormalisation ne dépendent pas de la dimension du système. Après avoir établi la maille élémentaire et une base pertinente d'opérateurs, j'ai programmé l'algorithme et j'ai étudié ses performances. Cet article a aussi une valeur pédagogique. En effet, les deux articles publiés à la maîtrise ne font que cinq pages chacun et la majorité des détails en sont absents. C'est pourquoi David et moi en avons profité pour expliciter tout ce qui est nécessaire à la programmation de l'algorithme

et pour décrire tous les concepts s'y rattachant : décodage tolérant aux fautes (version 3D du problème), mailles élémentaires, bases d'opérateurs et règles de passage de messages. Nous avons également ajouté une annexe expliquant comment prendre des marginales sur une distribution de probabilité sur le groupe de Pauli. Pour ces raisons, j'ai décidé de n'inclure que l'article directement.

7.2 Article

Fault-Tolerant Renormalization Group Decoder for Abelian Topological Codes

Guillaume Duclos-Cianci and David Poulin
Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1, Canada
(Dated: April 19, 2015)

We present a three-dimensional generalization of a renormalization group decoding algorithm for topological codes with Abelian anyonic excitations that we introduced for two dimensions in [1, 2]. This 3D implementation extends our previous 2D algorithm by incorporating a failure probability of the syndrome measurements, i.e., it enables fault-tolerant decoding. We report a fault-tolerant storage threshold of $\sim 1.9(4)\%$ for Kitaev's toric code subject to a 3D bit-flip channel (i.e. including imperfect syndrome measurements). This number is to be compared with the 2.9% value obtained via perfect matching [3]. The 3D generalization inherits many properties of the 2D algorithm, including a complexity linear in the space-time volume of the memory, which can be parallelized to logarithmic time.

I. INTRODUCTION

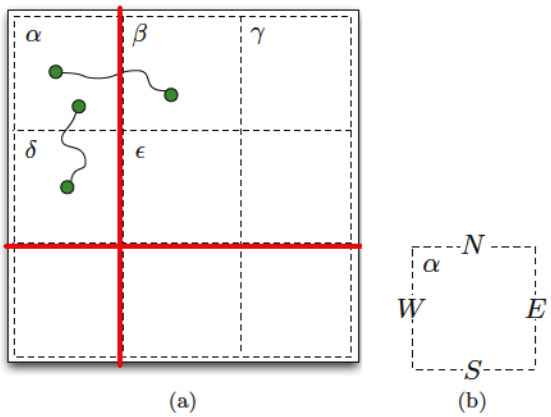
Topological quantum error-correcting codes currently stand as some of the most promising implementations of quantum memories and computers. Crudely, topological codes are standard quantum error-correcting codes with additional geometric constraints: their check operators involve only neighbouring spins on a two dimensional (2D) lattice. As a consequence, they can exhibit high fault-tolerant threshold [4–6] with relatively low overhead. Some topological codes also support transversal implementation of Clifford gates [7], which simplifies fault-tolerant quantum computation. Lastly, topological codes can be efficiently decoded [1, 3, 8], which is the topic of this paper.

Decoding a quantum code consists in inferring the optimal recovery given a statistical description of the noise and an error syndrome—i.e., the measurement outcome of check operators which reveal incomplete information about the particular error that has affected the system. Thus, decoding is a classical statistical inference problem involving a very large number of correlated random variables. Extremely fast decoding algorithms are required to prevent errors from building up in between error correction cycles, although some lag-time can be tolerated, e.g., by extending ideas from [9]. In [1, 2], we introduced a decoding algorithm for Kitaev's topological code [10] that uses renormalization group (RG) techniques from statistical physics. It's complexity is linear with the number of qubits, as compared to the cubic complexity of previously known algorithms [11]. Most importantly, it can be parallelized to logarithmic time.

The present paper is a continuation of our work initiated in [1, 2], and serves many purposes. 1) Our previous work focused on error correction in the presence of perfect syndrome measurements. When measurements are faulty, fault-tolerant techniques are required which change the nature of the decoding problem. As we explain below, for topological codes, this can be effectively described by increasing the lattice dimension by one dimension representing time [8]. Thus, we adapt our RG algorithm, initially devised for a 2D lattice, to a 3D fault-tolerant setting.[1] 2) Our algorithm was devised specifically for Kitaev's topological code. Because all 2D stabilizer codes are locally equivalent to multiple copies of Kitaev's code [13], our RG algorithm can be used with any such code. However, this requires determining the local mapping that realizes this equivalence, and transforming the local noise model accordingly, which can in principle affect the decoder's performances. Here, we describe our methods in physical terms that are directly applicable to any code that supports Abelian anyons [10, 13–16], not restricted to stabilizer codes. We have implemented a special case of this generalization in [16] for the $\mathbb{Z}_d$ quantum double model. 3) Our previous publications on this topic focused on applications, giving only a high level description of the actual algorithm. Here, we provide a complete detailed description of the structure of the algorithm, which should be sufficient for anyone interested in implementing it.

The rest of the paper is organized as follows. In the next section, we provide a heuristic physical description of the algorithm in terms of localized Abelian anyons. This section should provide a good physical intuition of the different components of the algorithm. This is first done assuming perfect syndrome measurements, and in the last subsection we explain how the problem is modified in the presence of faulty errors, following [8]. Section III revisits all the concepts introduced heuristically in Sec. II for the special case of Kitaev's topological code, using an algebraic formalism closely related to the actual implementation of the algorithm. Section IV presents our numerical experiments, and we conclude in Sec. V with possible extensions and relations to other methods. Appendix A details our mathematical notation for probability distributions over the n-qubit Pauli group.

[1] Note that we have used our algorithm in a fault-tolerant setting in [12], but did not provide any details of the implementation.



(a) (b)

FIG. 1: (a) A 2D topological code is cut into unit cells α, β, ... Gauge lines representing the non-trivial cycles (solid red lines) are chosen arbitrarily. Computing the flow of charge through the gauge lines is equivalent to decoding. (b) Each region has four boundaries that we label north (N), east (E), south (S) and west (W).

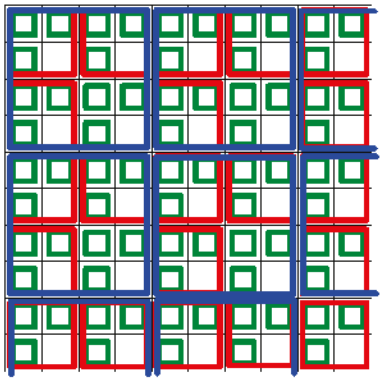


FIG. 2: Structure of the RG cells. A unit cell is composed of four regions (unit cells of the previous RG iteration). In each unit cell (red square), the charge of only three of the four regions is measured (green squares); the south-east corner is not measured, leaving the total charge of the unit cell undetermined. This missing measurement is replaced at the following RG iteration by a measurement of the entire unit cell (red square), which is now a region of a renormalized unit cell (blue square). Note that this modification of the charge measurement does not need to be implemented physically, it only reflects a change in bookkeeping.

II. HEURISTIC PHYSICAL DESCRIPTION

In this Section, we provide a heuristic physical description of the problem of interest, and of the numerical tools we have developed to solve it. A more detailed mathematical description is presented in Sec. III.

A. Decoding problem

Consider a 2D sheet of topological matter supporting Abelian anyons. For simplicity, suppose that the system has periodic boundary conditions, so it forms a torus. The information is encoded in the degenerate ground state of the system. Excitations above the ground state manifold are localized Abelian anyons—they carry conserved charges $\{a, b, c, \ldots\}$ that obey "deterministic" fusion rules, e.g. $a \times b = c$. The information in the ground state can be modified by creating a particle-antiparticle pair $(a, \bar{a})$, dragging one of the particle around a topologically non-trivial cycle, and fusing it with its original partner $a \times \bar{a} = 1$.

In the presence of errors, such a process could occur spontaneously. For instance, the creation of a particle-antiparticle pair could result from a thermal fluctuation. Once created, additional errors could cause the particles to diffuse on the sheet. To prevent corruption of the memory, we must therefore keep track of the homology of the particles' world-lines. Periodic measurements of the particles' location yield partial information about their trajectories, and the *decoding problem* becomes one of statistical inference: it sets to determine the most likely homology of the particles' world-lines given two consecutive snapshots of their locations. Concretely, we can arbitrarily choose two gauge lines representing the two non-trivial cycles of the torus [c.f. Fig. 1(a)], and the decoding problem consists in determining the net flow of charge, or current, across these two gauge lines.

B. RG algorithm

In [1, 2], we proposed a renormalization group technique to tackle this problem. First, we break the lattice into 2×2 sublattices, or "unit cells", as illustrated on Fig. 1(a). Given a microscopic noise model, we can compute the probability for the value of the current across each of the four walls [North, South, East, West, c.f. Fig. 1(b)] of each cell, conditioned on the charge configuration observed inside this cell. This produces a probability distribution $\mathcal{P}_\alpha(N_\alpha, E_\alpha, S_\alpha, W_\alpha)$ for each cell α, where $N_\alpha, E_\alpha, S_\alpha, W_\alpha$ take values representing the possible currents.[2]

Concretely, the presence of a charge, say, in the north-east corner of the unit cell would lead to the assignment of a probability $\mathcal{O}(p)$ to a current through the northern or eastern walls, and a probability $\mathcal{O}(p^2)$ for the southern or

[2] To specify the mathematical structure of the current variables, we can choose a minimal set $\{a_1, a_2, \ldots a_k\}$ of k "elementary" charges that generate all other charges under fusion. Then, any charge can be written as $a_1^{\alpha_1} \times a_2^{\alpha_2} \times \ldots a_k^{\alpha_k}$, or more succinctly represented by the vector $(\alpha_1, \alpha_2, \ldots, \alpha_k) \in \mathbb{Z}^{h_1} \times \mathbb{Z}^{h_2} \times \ldots \mathbb{Z}^{h_k}$ where h_j is the order of charge a_j, meaning that h_j copies of a_j always fuse to the identity. Then, the current variables N_A, S_A, E_A, and W_A each take value in $\mathbb{Z}^{h_1} \times \mathbb{Z}^{h_2} \times \ldots \mathbb{Z}^{h_k}$. In the case of the toric code for instance, there are two elementary charges, e and m, and their order is 2 since $e \times e = m \times m = 1$.

western walls, reflecting the fact that the first two cases require only one error process while the second two cases require two error processes. Here, p represents the probability of an error process such as particle creation, annihilation, or displacement. The big-$\mathcal{O}$ hides multiplicative factors accounting for the distinct error processes resulting in the same currents, as well as higher order processes. In any case, these probabilities can be computed exactly given an underlying local noise model.

After having computed these current probability distributions for every cell, we merge groups of four neighbouring unit cells into renormalized cells (c.f. Fig. 2) and iterate the procedure: we sum over all the bulk processes that lead to a given current across each of the four renormalized boundaries of each cell. This is done as explained above, except that the error probability p is not uniform on the lattice, but is given by the current variables of the previous RG iteration. By successive iteration, (and assuming for simplicity that the lattice linear dimension is a power of 2) we arrive at a situation where the Northern and Western walls actually correspond to the gauge line representing the non-trivial cycles of the torus. Determining the current across these walls is equivalent to decoding, as explained above.

The difficulty with the procedure we outlined above is that charge conservation imposes strong correlations between the current variables, so their exact joint probability cannot be computed efficiently. To see this, note that the current variables are subject to two constraints. (a) The sum of the current entering a cell must be equal to the total charge inside the region. This leads to a conservation equation $N_\alpha + S_\alpha + E_\alpha + W_\alpha = c_\alpha$ for each cell α, where c_α is the total charge contained in α, and is known from observation (error syndrome). (b) The currents associated to juxtaposed walls of neighbouring cells must be equal and opposite, e.g. $S_\alpha = -N_\delta$ when δ is the cell directly to the south of cell α, see Fig. 1(a). This simply follows from the fact that, e.g. S_α and N_δ are actually associated to the same physical boundary. Constraints (a) correlate the variables of a given cell while constraint (b) correlate variables between different cells, so the distribution is globally correlated.

Thus, approximations are required to solve this problem efficiently, as we now explain. First, just as a matter of bookkeeping, each cell stores only the random variables associated to its northern and western walls, the other ones are redundant from constraint (b). This does not affect the correlated nature of the problem however since (a) becomes $N_\alpha + W_\alpha - N_\delta - W_\beta = c_\alpha$ [c.f. Fig. 1(a)], and (b) now says that e.g. $\mathcal{P}_\alpha(N_\alpha, W_\alpha, N_\delta, W_\beta)$ and $\mathcal{P}_\beta(N_\beta, W_\beta, N_\epsilon, W_\gamma)$ must be the marginals of one global distribution $\mathcal{P}(N_\alpha, W_\alpha, N_\delta, N_\beta, W_\beta, N_\epsilon, W_\gamma)$. To simplify the problem, we relax this condition to a "mean-field" condition, demanding that the two distributions yield the same marginals along the wall they share, i.e. $\mathcal{P}_\alpha(W_\beta) = \mathcal{P}_\beta(W_\beta)$, where the marginals are defined the usual way

$$\mathcal{P}_\alpha(W_\beta) = \sum_{W_\alpha, N_\delta, N_\alpha} \mathcal{P}_\alpha(N_\alpha, W_\alpha, N_\delta, W_\beta) \qquad (1)$$

$$\mathcal{P}_\beta(W_\beta) = \sum_{N_\beta, N_\epsilon, W_\gamma} \mathcal{P}_\beta(N_\beta, W_\beta, N_\epsilon, W_\gamma). \qquad (2)$$

These mean-field conditions are enforced heuristically using belief propagation [17].

Since mean-field approximations are not reliable in strongly correlated systems, we make one more modification to the problem. Charge conservation imposes a hard constraint (a) to the current variables, which is unlikely to ever be fulfilled in a mean-field approximation. To circumvent this problem, we let the charge c_A inside each cell fluctuate, i.e., we treat it as a random variable. To describe this procedure, recall that each unit cell is composed of a collection of four regions (i.e. unit cells of the previous RG iteration). Measuring the charge distribution inside the unit cell amounts to measuring the total charge in each of these regions, which clearly fixes the total charge of the unit cell. In the modified procedure, we measure the charge of all but one of the regions, say the south-east region. As a consequence, the total charge of the unit cell is undetermined, which relaxes the constraints on the current variables as desired. This procedure is illustrated on Fig. 2. The charge of the unit cell is only fixed at the following RG iteration.

C. Fault-tolerant decoding

Our description of the problem so far assumes that the charge measurements are perfect. A realistic noise model would also include faulty measurements, i.e. every charge measurement has some probability of reporting the wrong charge. To alleviate this problem, measurements can be repeated in time. A different outcome between two consecutive measurements can then be caused either from an actual error having occurred in the time between the measurements—e.g. a particle has moved in this region—or by an error in one of the two measurements.

Consider the space-time cube enclosed between two consecutive local charge measurements (c.f. Fig. 3). We can associate a topological charge to this cube equal to the difference between the charges revealed by the two measurements enclosing it. If the charge of a cube is non-trivial, it means that the two consecutive measurements did not yield the same result. As explained above, this could be caused by a "space-like error" taking place between the two measurements, or a "time-like error" affecting the measurements themselves, see Fig. 3. In any case, the total current across the six walls of the cube must be equal to the charge of the cube. We then see [8] that the decoding problem becomes that of determining the world-line homology of the particles in space-time.

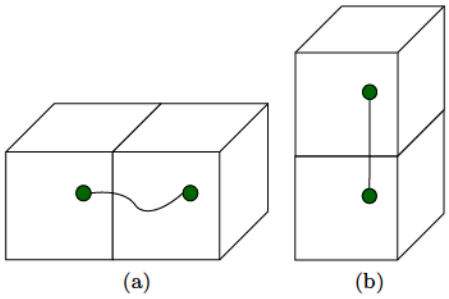


FIG. 3: Space-time diagram of the fault-tolerant error-correction procedure; time flows vertically (a) A space-like error is an error that affects a qubit in between two measurements. It creates excitations in the two cubic cells with which it overlaps. (b) A time-like error is caused by a faulty measurement. It creates excitations in the two cubic cells separated by that measurement.

Thus, the fault-tolerant decoding problem differs from the decoding problem with perfect measurements only in respect of the lattice dimension. Hence, the RG decoding algorithm outlined above can be applied directly.

III. FORMAL DESCRIPTION FOR KITAEV'S TORIC CODE

In this Section, we describe more rigorously the concepts introduced in the previous Section for the special case of Kitaev's toric code (KTC). We begin with the 2D scenario as it is technically simpler, yet conceptually equivalent to 3D. The system is a $\ell \times \ell$ square lattice, Λ, with periodic boundary conditions. We assume that ℓ is a integer power of 2. Each site $\Lambda_{i,j}$ $(0 \leq i,j < \ell)$ holds two qubits, $\Lambda_{i,j,\alpha}$ ($\alpha \in \{H,V\}$, where H and V stand for horizontal and vertical, respectively). The KTC on the torus is a stabilizer code [18] and we assume familiarity with this class of codes.

A. Model

The stabilizer group of KTC is generated by two types of operators. On every site, $\Lambda_{i,j}$, define a *site operator*, $A_{i,j} = X_{i,j,H}X_{i,j,V}X_{i,j-1,H}X_{i-1,j,V}$, and on every plaquette, define a *plaquette operator*, $B_{i,j} = Z_{i,j,H}Z_{i,j+1,V}Z_{i+1,j,H}Z_{i,j,V}$ (see Fig. 4). Let $S_g = \{A_{i,j}, B_{i,j}\}$ be the set of all plaquette and site operators. Note that it is invariant under translation. The codespace is defined to be the simultaneous +1 eigenspace of all the stabilizer operators. Equivalently, we can define the Hamiltonian $H = -\sum_{Q\in S_g} Q$, and the codespace is the degenerate ground space of H. There are $n = 2\ell^2$ qubits on the lattice but only $2\ell^2 - 2$ independent generators, i.e. S_g is overcomplete. Indeed, one can easily verify that the stabilizer generators obey the

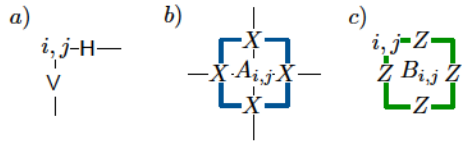


FIG. 4: a) One site, $\Lambda_{i,j}$, of the square lattice, Λ, on which is defined KTC. Qubits, $\Lambda_{i,j,0}$ and $\Lambda_{i,j,1}$, live on the edges and are associated to sites with the convention depicted. b) Site operator $A_{i,j} = X_{i,j,H}X_{i,j,V}X_{i,j-1,H}X_{i-1,j,V}$. Blue strings represent X operators. c) Plaquette operator $B_{i,j} = Z_{i,j,H}Z_{i,j+1,V}Z_{i+1,j,H}Z_{i,j,V}$. Green strings represent Z operators.

two global constraints $\prod_{i,j} A_{i,j} = \mathbb{1}$ and $\prod_{i,j} B_{i,j} = \mathbb{1}$. This implies that two logical qubits are encoded in the codespace.

The logical X and Z operators acting on the encoded qubits are non-trivial homological cycles (i.e loops around the torus) of X operators on the dual lattice and Z operators on the direct lattice. We arbitrarily choose the *bare logical* operators to be

$$\overline{Z}_0 = \prod_j Z_{0,j,H} \qquad \overline{Z}_1 = \prod_i Z_{i,0,V} \qquad (3)$$
$$\overline{X}_0 = \prod_i X_{i,\ell-1,H} \qquad \overline{X}_1 = \prod_j X_{\ell-1,j,V}.$$

These correspond to the gauge lines introduced in the previous section, c.f. Fig. 1(a).

Errors are modeled by random Pauli operators affecting the qubits. A Pauli operator will in general anti-commute with a subset of the elements of S_g, causing their eigenvalues to flip from +1 in the codespace to -1. An element of S_g with -1 eigenvalue corresponds to a local excitation, an Abelian anyon. We refer to a plaquette excitation as a magnetic flux and to a site excitation as an electric charge. It is useful to associate binary matrices, $\mathbf{a}_{i,j}$ ($\mathbf{b}_{i,j}$) to an excitation configuration, with entries 0 if the eigenvalue of $A_{i,j}$ ($B_{i,j}$) is +1 and entries 1 otherwise. Thus, the excitation configuration associated to the product of two errors is the binary sum of their respective excitation configurations—the two distinct topological charges are their own inverse.

Since the Pauli operator $X_{i,j,H}$ anti-commutes with plaquettes $(i-1,j)$ and (i,j), we see that X operators can create a pair of magnetic fluxes, move a magnetic flux to a neighbouring plaquette, and annihilate a pair of neighbouring magnetic fluxes. The Z Pauli operator plays an equivalent role for electric charges. Thus, the microscopic noise model describing the dynamics of the anyons can be specified by a memoryless Pauli channel $\mathcal{P}_{i,j,\alpha}(Q)$, $Q \in \{I, X, Z, Y = iXZ\}$—i.e., a probability distribution over the four Pauli operators for each qubit of the lattice. In this model, the errors E affecting the system are thus elements of the n-qubit Pauli group $\mathcal{G}^n$. The probability of an error $E = \bigotimes_{i,j,\alpha} Q_{i,j,\alpha}$ is simply given by $\mathcal{P}(E) = \prod_{i,j,\alpha} \mathcal{P}_{i,j,\alpha}(Q_{i,j,\alpha})$.

B. Decoding problem

When an error $E \in \mathcal{G}^n$ affects the system initially i codespace, the task of error-correction is to bring the sy tem back in the codespace by matching every excitatio in pairs—thus annihilating them all—without changir the encoded information. This is realized by applyir a correction operator, $C \in \mathcal{G}^n$. If the total operat EC is homologically non-trivial, a logical operation wi be implemented as the system is brought back to t codespace, so the information will be corrupted. To k successful, the correction C must therefore be homolog cally equivalent to the error E.

The decoding problem can be formulated in terms this equivalence. Given an error syndrome—i.e., an exc tation configuration—the decoder must find a Pauli operator that is homologically equivalent to the error that has created this syndrome. This is a statistical inference problem. One approach to this problem is to find, among all errors that are consistent with the observed excitation configuration, the one with the highest probability. When the noise model is independent and uniform, this error is simply the lowest weight operator consistent with the excitation configuration, where the weight of C is the number of non-trivial single-qubit Pauli operators in C. The Perfect Matching Algorithm (PMA) performs this task with a $\mathcal{O}(\ell^6)$ complexity [3, 8].

This turns out not to be the optimal solution however. To understand this, let t denote an operator with the correct excitation configuration. We suppose that t is chosen in some canonical way, so it is in one-to-one correspondence with excitation configurations. The probability that the error E is homologically equivalent to t is simply proportional to the sum of the error probability $\mathcal{P}(Q)$ over all errors Q equivalent to t. Since the equivalence relation is generated by elements of the stabilizer group $\mathcal{S}$, this is $\sum_{s\in\mathcal{S}} \mathcal{P}(ts)$. On the other hand, t could differ from the actual error E by a combination of logical operators Eq. (3), i.e. a non-trivial cycle. Thus, we can use the group generated by the logical operators Eq. (3) to label the equivalence classes of errors. Generalizing the above reasoning, the probability that the error E is homologically equivalent to tl defines the probability associated to the class $l \in \langle \overline{X}_i, \overline{Z}_i \rangle$ conditioned on the excitation configuration (or equivalently conditioned on t):

$$\mathcal{P}(l|t) = \frac{1}{\mathcal{P}(t)} \sum_{s\in\mathcal{S}} \mathcal{P}(tls) \tag{4}$$

where the normalization factor is $\mathcal{P}(t) = \sum_{l,s} \mathcal{P}(tls)$. The optimal decoding consists in choosing the l that maximizes Eq. (4) (so the normalization $\mathcal{P}(t)$ is not relevant). The product tls is a specific Pauli operator and $\mathcal{P}(tls)$ is the probability of this operator as given by the noise model. This computation is intractable because $|\mathcal{S}|$ scales exponentially with the system size.

The type of mathematical manipulation leading to Eq. (4) will be used extensively by the algorithm and in the following discussion, so Appendix A provides some formal background and examples that should be consulted before reading the next sections.

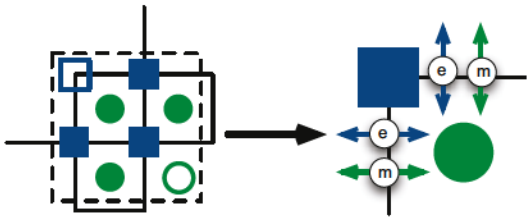


FIG. 5: Left: Choice of a 2×2 unit cell used to perfom the RG on the KTC. Green disks represent plaquette operators, blue squares represent site operators, and edges represent qubits. The two generators which are left out are represented by an empty square and a circle. Right: The RG yields a renormalized lattice. The eigenvalue of the renormalized generators corresponds to the total charge of the region and the renormalized noise model corresponds to the net flow of charges throught the boundaries (eq. 6).

C. RG decoding algorithm

The RG algorithm decomposes the lattice into unit cells. We choose them to be 2×2 squares enclosing four plaquette and four site generators, see Fig. 5. As explained in the previous section, the RG decoder requires knowledge of all but one of the magnetic and one of the electric operators it encloses. By symmetry, we choose to leave out the south-east plaquette operator and the north-west site operator. As a consequence, the scheme will follow our description of Sec. II for the magnetic fluxes, but for the electric charges the lattice is rotated by 180^o relative to our description of Sec. II . We include in the cell all the qubits that participate in the excitations measured operators, so a cell contains 12 qubits in total. Some of the qubits are shared between neighbouring cells, and this will be responsible for the constraint (b) that correlates their current variables.

To set up calculations, we define the following basis for the 12 qubits of the unit cell, see Fig. 6 for qubit labeling

$$\begin{array}{llll}
S_0 = X_0X_2X_3X_8 & T_0 = Z_0 & E_0 = X_6X_{10} & \overline{X}_0 = X_2X_6 \\
S_1 = X_1X_4X_5X_9 & T_1 = Z_1 & E_1 = X_7X_{11} & \overline{X}_1 = X_5X_7 \\
S_2 = X_3X_4X_6X_7 & T_2 = Z_0Z_3 & E_2 = Z_0Z_8 & \overline{Z}_0 = Z_0Z_2 \\
S_3 = Z_0Z_1Z_3Z_4 & T_3 = X_4X_7 & E_3 = Z_1Z_9 & \overline{Z}_1 = Z_1Z_5 \\
S_4 = Z_2Z_3Z_6Z_{10} & T_4 = X_6 & E_4 = X_8 & \\
S_5 = Z_4Z_5Z_7Z_{11} & T_5 = X_7 & E_5 = X_9 & \\
 & & E_6 = Z_{10} & \\
 & & E_7 = Z_{11} &
\end{array} \tag{5}$$

The physical interpretations of these operators are the following. The stabilizer generators S_j are the six excitation measurement operators used in the unit cell; they are plaquette and site operators. The T_j are the

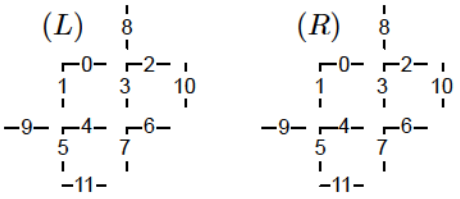


FIG. 6: Two neighbouring unit cells labeled L and R. Each shows the labeling of the qubits used in Eq. (5). Note that since these two cells are neighbours, they share qubits. In particular, qubits 6 and 10 in cell L are the same as qubits 9 and 1, respectively, in cell R.

associated canonical pure errors in the sense that $t = T_0^{a_{i,j+1}} T_1^{a_{i+1,j}} T_2^{a_{i+1,j+1}} T_3^{b_{i,j}} T_4^{b_{i,j+1}} T_5^{b_{i+1,j}}$ produces the excitation configuration $\mathbf{a}_{i,j}$ $\mathbf{b}_{i,j}$ inside the cell, without inducing any magnetic flow through the northern or western wall or any electric flow through the southern or eastern wall. The logical operators $\overline{X}_i$ and $\overline{Z}_i$ monitor respectively the magnetic current across the north ($i = 0$) and west ($i = 1$) wall and the electric current across the east ($i = 0$) and south ($i = 1$) wall. Thus, they correspond to the current variables used in Sec. II. Lastly, the E_j are errors that change the charge of the site and plaquette operators that have been left out of the cell. For instance, E_0 brings a magnetic flux through the eastern wall into the south-east corner.

An RG iteration takes an excitation configuration and a probability distribution over the Pauli group of the qubits contained inside the unit cell, and outputs a current probability distribution obtained by summing over all equivalent processes that are consistent with the observed excitation configuration. For example, the operator X_0 (see Fig. 6 for labeling) is equivalent to the operator X_2X_3 as it corresponds to a flow of one magnetic flux through the north boundary and into the north-west plaquette. This is more directly seen when decomposed in the basis Eq. (5): $X_0 = T_3\overline{X}_0 S_0 S_2 E_4$ and $X_2X_3 = T_3\overline{X}_0 S_2$ since both decompose into the logical operator $\overline{X}_0$, which is associated to the magnetic current through the northern wall, and the pure error T_3 which is conjugate to S_3, the north-west plaquette. Thus, if a magnetic flux was indeed observed in the north-west corner, both of these errors should contribute to the probability of a magnetic flow through the northen boundary. More generally, the probability of a current $l \in \langle \overline{X}_i, \overline{Z}_i \rangle$ conditioned on a charge configuration $t = T_0^{a_{i,j+1}} T_1^{a_{i+1,j}} T_2^{a_{i+1,j+1}} T_3^{b_{i,j}} T_4^{b_{i,j+1}} T_5^{b_{i+1,j}}$ is given by

$$\mathcal{P}(l|t) \propto \sum_{s,e} \mathcal{P}(tles) \tag{6}$$

where $s \in \langle S_i \rangle$ relates topologically equivalent trajectories and $e \in \langle E_i \rangle$ changes the value of the undetermined charge, and we left out the normalization factor $\mathcal{P}(t)$.

D. Belief propagation

In the unit cell of Fig. 6 we see that there are eight qubits that belong to two unit cells; they are labeled $0, 1, 6, 7, 8, 9, 10$ and 11. For instance, qubit 1 of cell R is the same as qubit 10 of cell L immediately to its left. As for any other qubits, knowledge of the excitation configuration affects the error probability of these qubits. For instance, suppose that the system contains only two magnetic fluxes, one in the north-east corner of cell L and one in the north-west corner of cell R. In cell L, the presence of the magnetic flux should yield a high probability of X error on qubits 2 and 10. In cell R, the presence of the magnetic flux should yield a high probability of X error on qubits 1 and 0. But since qubits 10 of cell L and 1 of cell R are actually the same, this charge configuration should globally result in a very peaked probability of an X error on that qubit: it sits in between the two magnetic fluxes. But locally, given only knowledge of the charge configuration on a unique cell, this conclusion cannot be reached.

More generally, given a probability distribution over the Pauli group of the unit cell $\mathcal{P}(tles)$, we can compute the marginal error probability $\mathcal{P}_q(tles|_q)$ for each qubit q, obtained by taking a marginal of $\mathcal{P}(tles)$ (c.f. App. A).[3] When a qubit is shared between two cells, e.g. such as in the above example, its marginal conditional distributions obtained from different cells will typically differ. This is a manifestation of a violation of constraint (b) described in Sec. II. As explained there, the exact solution would be to demand that the conditional probability distribution assigned by each cell be consistent with one global probability distribution. Because of global correlations this problem is intractable, so we settle for a relaxed condition that the marginal probability distributions all agree.

This condition is enforced by a belief propagation algorithm. This is a local message passing algorithm where messages are exchanged between neighbouring cells, there is one message per shared qubit. Initially, the outgoing messages at a cell $m_q^{\text{out}}(p)$ are equal to $\mathcal{P}_q(tles|_q)$ computed in that cell. These outgoing messages are then exchanged between neighbouring cells, and become incoming messages, e.g. a cell L sends to its right neighbour R the message m_1^{out} that becomes m_{10}^{in} in R, and receives from R the message m_{10}^{out} that becomes m_1^{in} in L. Subsequent rounds of messages can be calculated using the received messages, following the prescription

$$m_q^{\text{out}}(p) \leftarrow \sum_{l,s,e} \delta(tles|_q, p) \frac{\mathcal{P}(tles)}{\mathcal{P}_q(tles|_q)} \prod_{q' \neq q} m_{q'}^{\text{in}}(tles|_{q'}), \tag{7}$$

[3] The base error prior is independent on each qubit, in which case this marginal consists in the noise model on qubit q. But because the RG can create a correlated noise model inside a unit cell, we need this more sophisticated notion of marginal, see App. A.

Here, $q, q' \in \{0,1,6,7,8,9,10,11\}$, $tles|_q$ is the restriction to qubit q of the Pauli operator $tles$, and $\mathcal{P}_q$ is the marginal on qubit q of the noise model as above (c.f. App. A). BP can be iterated a few times (e.g. three rounds) before executing a RG step. The messages are used to update the prior error probability, effectively replacing Eq. (6) by

$$\mathcal{P}(l|t) \propto \sum_{e\in\langle E_0,E_1\rangle} \sum_{s\in\langle S_0S_1S_2\rangle} \mathcal{P}(tles) \prod_q m_q^{\text{in}}(tles|_q). \quad (8)$$

E. Fault-tolerant decoding

The prescription given for the 2D decoding problem can be applied relatively straightforwardly to the 3D problem arising from fault-tolerant decoding in the presence of faulty syndromes. To simplify the description, we will assume that there are only bit-flip errors (X errors), so we only need to consider magnetic fluxes. The exact same method applies to Z errors and electric charges, and moreover both types of errors can be considered simultaneously (including Y errors).

We label by $0 \leq k < \tau$ the time at which the charge measurements are performed, where τ is the total duration of the computation (e.g. here we typically set $\tau = \ell$ to obtain a space-time cube). Errors affect the qubits in between measurements, and we use the label k for an event that occurs in between measurement $k-1$ and k. There are now two types of errors to be considered. Space-like errors $\boldsymbol{\eta}^k$ ($\eta^k_{i,j,\alpha} \in \mathbb{Z}_2$) result in the application of the Pauli operator $E^k = \prod_{i,j,\alpha} X^{\eta^k_{i,j,\alpha}}$ to the qubits between measurements $k-1$ and k. Time-like errors $\boldsymbol{\mu}^k$ ($\mu^k_{i,j} \in \mathbb{Z}_2$) result in inverting the measurement outcome at space-time coordinate (i,j,k) when $\mu^k_{i,j} = 1$.

The excitation configuration measured at time k results from the accumulation of space-like errors at times prior or equal to k, plus the measurement errors at that time, i.e. $\mathbf{b}^k = \boldsymbol{\mu}^k + \text{conf}(\prod_{k'\leq k} E^{k'}) = \boldsymbol{\mu}^k + \sum_{k'\leq k} \text{conf}(E^{k'})$. Thus, the difference between two consecutive rounds of measurements is $\Delta\mathbf{b}^k \equiv \mathbf{b}^{k-1} + \mathbf{b}^k = \boldsymbol{\mu}^{k-1} + \boldsymbol{\mu}^k + \text{conf}(E^k)$. In other words, $\Delta b^k_{i,j} = \mu^{k-1}_{i,j} + \mu^k_{i,j} + \eta^k_{i,j,H} + \eta^k_{i,j,V} + \eta^k_{i+1,j,H} + \eta^k_{i,j+1,V}$. This defines a local space-time cubic check operator.

In this 3D picture, a $\Delta b^t_{i,j} = 1$ plays the role of a magnetic flux. Note that each single error—either spatial or temporal—creates a pair of fluxes. In particular, the set of all errors can be viewed as a product of strings with magnetic fluxes located at their endpoints.

To formalize this description, define a 3D cubic lattice of bits, Λ, with sites $\Lambda_{i,j,k}$, holding three bits, $\Lambda_{i,j,k,\alpha}$ ($\alpha \in \{H,V,T\}$) with the convention that bits live on faces (see Fig. 7). The *error history*, E, on the 3D lattice is the combination of all space-like errors $\boldsymbol{\eta}$ and time-like errors $\boldsymbol{\mu}$, i.e. $E_{i,j,k,\alpha} = \eta^k_{i,j,\alpha}$ ($\alpha \in \{H,V\}$) and $E_{i,j,k,T} = \mu^k_{i,j}$. The excitation configuration associated

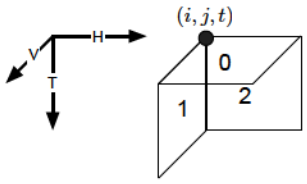


FIG. 7: Convention chosen for axis and unit cell of the 3D cubic lattice Λ. Bits are located on faces.

to E is $\Delta b^k_{i,j}$. In the following, we consider periodic boundary conditions in the spatial dimension to simplify the presentation. Then, as in the 2D case, two error histories are equivalent if they have the same excitation configuration and their product is homologically trivial on the three-torus.

The decoding problem thus stays qualitatively the same: find the most likely equivalence class of error histories consistent with the error syndrome. One subtle difference has to do with homologically non-trivial time-like loops, which do not carry the same physical meaning as space-like homologically non-trivial loops (logical operations). This difference is only caused by the unphysical boundary conditions that were chosen to simplify the presentation and the numerical simulations, and would not occur with open boundaries. In any case, a time-like logical error should not be regarded as a true memory corruption.

As in 2D, perfect matching [3] can be used to solve an approximate version of this problem, that consists of finding, among all error histories consistent with the excitation configuration, the one with highest probability.

The optimal solution however consists in finding the most likely equivalent class of errors, and this problem can be approximated with RG techniques. The RG decoding has the same logical structure as in 2D. The lattice is broken into $2\times2\times2$ unit cells. Each of these unit cells contain eight check operators (one of which is left undetermined) and 33 qubits, nine of which are shared. The current distribution over the three walls H, V, and T are computed by summing over the bulk configurations consistent with a given current and excitation configuration.

There is obviously a computational cost associated to summing over the bulk processes of a unit cell. This cost grows exponentially with the number of qubits contained inside the cell. For this reason, decoding a $2\times2\times2$ unit cell involves summing over 26-bit configurations (the cell contains 33 qubits and has seven check constraints), which is fairly demanding. For this reason, we choose to work with smaller unit cells.

To make the renormalization method for fault-tolerance practical, we consider asymmetric decoding. The simplest unit cell has dimensions $2\times1\times1$ (see Fig. 8). In this case, the cell contains one magnetic flux operator and renormalizes only one dimension of the lattice: $\ell\times\ell\times\ell \to \ell/2\times\ell\times\ell$. For the next step, rotate the cell to renormalize another direction, e.g. $\ell/2\times\ell\times\ell \to \ell/2\times\ell/2\times\ell$. Finally, consider a second rotation to renor-

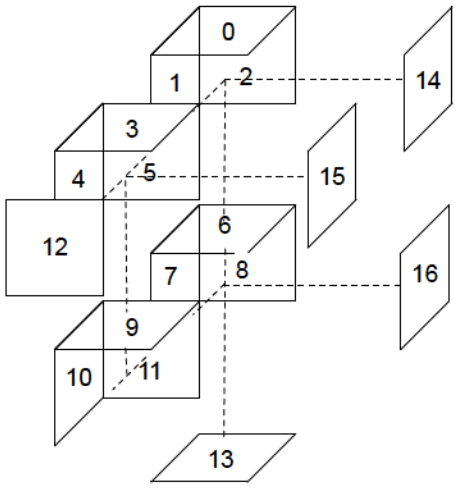


FIG. 9: Exploded view of a $2 \times 2 \times 1$ unit cell. Qubits 0, 1, 2, 4, 7, 12, 13, 14, 15 and 16 are shared. See Eq. (10) for the operator basis.

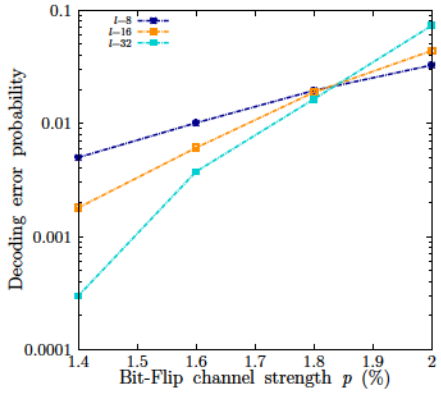


FIG. 10: Monte Carlo estimation of the decoding error probability as a function of bit-flip channel strength, p using a $2 \times 1 \times 1$ unit cell. A threshold is observed at $\sim 1.82\%$ (sample size: 10^4 per point).

values should be compared to the 2.9% value obtained via PMA [3] with the same error model.

Note that the $2 \times 2 \times 1$ unit cell is only compatible with lattice sizes that are powers of four. Moreover, due to the large size of the unit cell, decoding is relatively slow in this case, which limits us to small lattices $\ell = 16$ and $\ell = 64$ in practice.[4] The crossing point of the corresponding two curves gives us little confidence that we have correctly identified the threshold. For this reason, we also simulated lattice sizes $\ell = 8$ and $\ell = 32$ using an hybrid techniques where the $2 \times 2 \times 1$ cell was used until

[4] The complexity of the RG scheme is proportional to the space-time volume of the lattice, while the complexity of PMA scales with the cube of this volume. However, the constant factor of the RG scheme is exponential with the volume of each unit cell. Although this is independent of the lattice size, the constant can be quite prohibitive for large unit cells. Note also that RG can be straightforwardly parallelized to run in time logarithmic with the space-time volume of the lattice, but we have not implemented this parallel version.

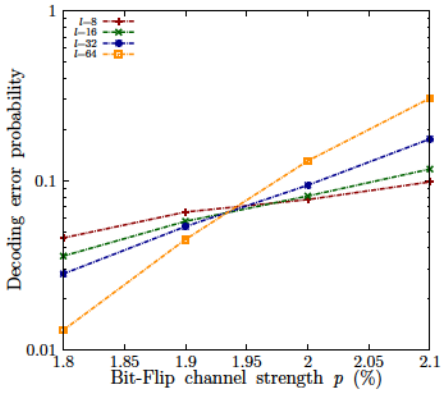


FIG. 11: Monte Carlo estimation of the decoding error probability as a function of bit-flip channel strength, p using a $2 \times 2 \times 1$ unit cell. A threshold is observed at $\sim 1.9(4)\%$. We note some finite size effects for $\ell = 8$, but not for the other three curves. Sample size varies from 3×10^3 to 10^4.

the very last step, where a $2 \times 1 \times 1$ call was used. The crossing point of all four curves agrees very well. This is not surprising since below threshold, we expect the error model to flow to a noiseless fixed-point, and therefore the failure rate should be largely independent of how decoding is performed at the last few RG iterations—the first RG iterations are the critical ones in determining the threshold. This observation also gives us confidence that RG could handle various lattice shapes by combining different unit cell shapes in the appropriate way deep in the RG flow.

One might suspect the threshold to be anisotropic—given the asymmetry in the RG, e.g. the direction that is renormalized first might exhibit a lower threshold. We analyzed the data by looking at the marginal error rate in the three different directions and saw no significant anisotropy. It is possible that the threshold is insensitive to such details, but that they have more subtle effect such as leading to different scaling exponents. In both cases, better statistics would be needed to give a quantitative answer.

V. CONCLUSION AND OUTLOOK

We have given a detailed presentation of a renormalization group algorithm for fault-tolerant decoding of topological quantum error-correcting codes supporting Abelian anyonic excitations. This extends our previous work [1, 2] in an essential way, permitting error correction in the presence of faulty measurements. We have numerically benchmarked this algorithm and found that it achieves a fault-tolerant error threshold of nearly 2%, in the same ballpark as the other leading techniques.

A. Relation to other work

Since the publication of our algorithm [1, 2], there has been a number of decoding algorithms proposed for topological codes that we now briefly review.

Sarvepalli and Raussendorf (SR) [19] have conceived a RG decoder for topological color codes that resembles ours in many ways. To our understanding their algorithm is conceptually identical to ours. Their presentation differs in one central way. Because some stabilizer generators unavoidably overlap with two different cells we were forced to share qubits between unit cells, which led to inter-cell correlations. Instead of this, SR split those stabilizer generators into two parts, each supported on a unique cell, and assign a binary random variable to the value of each part. The sum of these two random variables must equal the value of the syndrome associated to the stabilizer. These auxiliary binary variables play a role analogous to the shared qubits in our description. For the color code, their decoder achieves a threshold of 7.8%, compared to 8.7% achieved by mapping the code to multiple copies of KTC and decoding them with our RG algorithm [13].

Bravyi and Haah (BH) [20] have proposed a RG decoder suitable for any topological code supporting localized Abelian anyons. It crucially differs from our approach by being based on *hard decisions*, while our approach uses *soft decisions*. In other words, the optimal recovery is only decided at the very last step of our RG iterations. At intermediate iterations, probabilities are assigned to various recoveries, but none of the options is ever ruled out until the very end. In contrast, in the BH scheme, decisions are taken to fuse certain pairs of excitations at intermediate iterations of the RG scheme. Hard decoders are conceptually simpler, and so lend themselves to more rigorous analysis. Indeed, BH were able to prove that their decoder achieves a finite threshold, while we can only provide numerical evidences for our algorithm. On the other hand, it is well known in classical coding theory that soft decoders achieve better performances [21]. In the quantum setting, it has been shown that soft decoder can achieve a higher threshold and greater noise suppression below threshold [22]. Their algorithm achieves a threshold of 6.7%.

Wootton and Loss (WL) [23] used Monte Carlo sampling to estimate the sum in Eq. (4), thus directly estimating the probability of each equivalence class of errors conditioned on the error syndrome. Since Monte Carlo is exact within statistical error, given a sufficiently large sample, this technique is optimal and consequently outperforms all other decoding algorithms. Indeed, they achieve a threshold of 18.5%, compared to 16.4% using our method with the same noise model. Its main drawback is that it is very slow compared to other methods, its runtime scales (morally) exponentially with the lattice size.

Lastly, Fowler, Whiteside and Hollenberg (FWH) [24] have implemented a parallelized version of Edmonds' per-

fect matching algorithm [11] (PMA), which was the first algorithm used to decode topological code [8]. This implementation runs in constant average time without any performance loss compared to the original PMA. Our understanding of this algorithm is that it is of Las Vegas type, meaning that its run-time is not pre-determined. For instance, in this parallel implementation, it is possible that one node of the cluster requires more time than other nodes. On a very large lattice, these fluctuations could become important, i.e. the probability that at least one node takes a time superior or equal to any finite T approaches one. Thus, care must be taken in the interpretation of this constant average runtime.

B. Extensions

It is possible to combine these techniques in various ways to obtain tradeoffs between runtime and error correction. For instance, the RG algorithm of BH can conceptually be seen as a degradation of our algorithm where probabilities on current variables $\mathcal{P}(l)$ are rounded up to the closest binary distribution

$$\mathcal{P}'(l) = \begin{cases} 1 & \text{if } l \text{ maximizes } \mathcal{P}(l) \\ 0 & \text{otherwise} \end{cases} . \tag{11}$$

Because of this simplicity, it is much faster than our algorithm. There exist intermediate degradations that could interpolate between these two extreme schemes. For instance, we could round up the distribution to the closest trinary distribution

$$\mathcal{P}'(l) = \begin{cases} 1 & \text{if } \mathcal{P}(l) \geq 1-\epsilon \\ 0 & \text{if } \mathcal{P}(l) \leq \epsilon \\ F & \text{otherwise} \end{cases} \tag{12}$$

where the flag symbol F is used to signal a potential error. Such a scheme was used by Knill [25] in the context of concatenated codes, which can be seen as a degradation of the scheme of [22] that uses the exact probability distribution. One could also consider keeping only the first few largest probabilities, and rounding all other to zero.

As we have seen, larger unit cells lead to better error correction, but the exact summation Eq. (6) of bulk processes inside a unit cell scales exponentially with the volume of the cell. One possibility would be to sum the bulk processes inside the unit cell only approximately. This would enable RG decoding using much larger unit cells. For instance, we could use WL's Monte Carlo's scheme to estimate this sum. Alternatively, we could use tensor-network techniques [26] to approximate this sum. Even without approximations, a transfer matrix approach could be used to decrease this complexity from exponential in the area of the cell (or volume in 3D) to exponential in its linear size (or area in 3D). For the small cells we considered here, these more elaborate techniques are of no use.

Lastly, we note that the description of our algorithm presented in Sec. II applies equally well to subsystem codes [27] that have local stabilizer generators in 2D, such as the topological subsystem color codes [15] (but excludes e.g. Bacon-Shor codes [28]). Indeed, the stabilizer generators of these codes reveal excitations that carry topological charges and the decoding problem consists of inferring the world-line homology of these excitations. The main difference is that not all topological charges can corrupt the encoded information. Some of the topological charges—that we called *gauge charges* in [13]—can be dragged along a non-trivial cycle without changing the ground state of the system. Thus, the current associated to these charges does not need to be monitored. Thus, Eq. (6) should contain an extra sum corresponding these harmless processes. We have used this technique for the topological subsystem color code in [13] and obtained a threshold of 1.95%.

C. Acknowledgements

We would like to thank Arvin Faruque for useful discussions. Computational resources were provided by Calcul Québec and Compute Canada. This work was partially funded by NSERC and by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract D11PC20167. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

Appendix A: Manipulating probabilities over $\mathcal{G}^n$

In this Appendix we provide some mathematical background for manipulating probabilities over the n qubit Pauli group $\mathcal{G}^n$. This should be useful to understand the details of Sec. III or to implement the RG algorithm.

Let $\mathcal{P}(E)$ be a probability distribution over the n-qubit Pauli group $\mathcal{G}^n$, e.g. corresponding to a physical noise model. Given a generating set $\{Q_i\}$ of $\mathcal{G}^n$, we can express any $E \in \mathcal{G}^n$ as

$$E = \prod_{i=1}^{2n} Q_i^{x_i} \tag{A1}$$

where $x_i \in \{0,1\}$. This allows us to interpret $\mathcal{P}(E)$ as a distribution over $2n$ binary variables $\mathcal{P}(x_1,...,x_{2n}) = \mathcal{P}(E = \prod_{i=1}^{2n} Q_i^{x_i})$. Standard Bayesian calculus can then be used to define marginal distributions, conditional distributions, etc. For instance, the marginal distribution over x_1, x_2, and x_3 is given by $\mathcal{P}(x_1, x_2, x_3) =$

$\sum_{x_4,\ldots x_{2n}} P(x_1, x_2, \ldots x_{2n})$. The probability of x_1 and x_2 conditioned on x_3 is given by $\mathcal{P}(x_1, x_2|x_3) = \mathcal{P}(x_1, x_2, x_3)/\mathcal{P}(x_3)$. These probabilities implicitly depend on a basis choice $\{Q_i\}$, and we can perform such manipulations for any basis of $\mathcal{G}^n$.

These definitions extend straightforwardly to more variables. With the isomorphism Eq. (A1), we can relabel these probabilities $\mathcal{P}(Q_1, Q_2, Q_3) = \mathcal{P}(x_1, x_2, x_3)$, and so forth.

We can create coarse grained variables associated to subgroups of the Pauli group. For instance, let $\mathcal{K} = \langle Q_1, Q_2, Q_3\rangle$ and $\mathcal{T} = \langle Q_4, Q_5\rangle$ be two subgroups of $\mathcal{G}^n$. An element K of $\mathcal{K}$ can be decomposed as $K = Q_1^{x_1} Q_2^{x_2} Q_3^{x_3}$, and similarly an element T of $\mathcal{T}$ can be decomposed as $K = Q_4^{x_4} Q_5^{x_5}$. The joint, marginal, and conditional probabilities can then be defined in a natural way

$$\mathcal{P}(K, T) = \mathcal{P}(KT) = \mathcal{P}(x_1, x_2, x_3, x_4, x_5) \quad \text{(A2)}$$
$$\mathcal{P}(K) = \mathcal{P}(x_1, x_2, x_3) \quad \text{(A3)}$$
$$\mathcal{P}(T) = \mathcal{P}(x_4, x_5) \quad \text{(A4)}$$
$$\mathcal{P}(K|T) = \mathcal{P}(x_1, x_2, x_3|x_4, x_5). \quad \text{(A5)}$$

These are the formal definitions behind Eqs. (4,6,7).

Lastly, we can convert any of these probabilities—joint, marginal, and conditional—to a different basis. For instance, let $\langle Q'_1, Q'_2, Q'_3\rangle$ be a different generating set for $\mathcal{K}$. We can express these generators in terms of the previous ones $Q'_i = \prod_{j=1,2,3} Q_j^{y_{ij}}$ with $y_{ij} \in \{0, 1\}$. Suppose that we have computed $\mathcal{P}(K|T) = \mathcal{P}(x_1, x_2, x_3|x_4, x_5)$ using the basis $\{Q_i\}$, and now wish to compute $\mathcal{P}(K|T)$ for $K = Q_1'^{z_1} Q_2'^{z_2} Q_3'^{z_3}$. Since

$$K = \prod_{i=1,2,3} \Big(\prod_{j=1,2,3} Q_j^{y_{ij}} \Big)^{z_i} \quad \text{(A6)}$$
$$= \prod_{j=1,2,3} Q_j^{\sum_{i=1,2,3} y_{ij} z_i}, \quad \text{(A7)}$$

we see that $\mathcal{P}(K|T) = \mathcal{P}(z_1, z_2, z_3|x_4, x_5) = \mathcal{P}(x_1, x_2, x_3|x_4, x_5)$ for $x_j = \sum_{i=1,2,3} y_{ij} z_i$. These probabilities can then be used to compute marginals over a subgroup of $\mathcal{K}$ specified in terms of the primed generators. For instance, for $F \in \langle Q'_1, Q'_2\rangle$ we have $\mathcal{P}(F|T) = \mathcal{P}(z_1, z_2|x_4, x_5)$. Thus, we see the usefulness of performing basis changes: it is used to adapt the probability to the particular subgroup we are interested in.

We will be using this type of manipulation in the special case where the basis $\{Q'_j\}$ actually corresponds to the basis of single qubit Pauli operators $\{X_i, Z_i\}$. In that case, for $K = \prod_i X_i^{\alpha_j} Z_i^{\beta_i}$ we will be using the special notation $K|_q$ to represent $X_q^{\alpha_q} Z_q^{\beta_q}$, i.e. the Pauli operator on qubit q in K. These are the formal definitions behind many mathematical expressions of Subsection III D.

[1] G. Duclos-Cianci and D. Poulin, Physical Review Letters **104**, 050504 (2009).
[2] G. Duclos-Cianci and D. Poulin, Information Theory Workshop (ITW) p. 1 (2010).
[3] J. W. Harrington, Ph.D. thesis, California Insitute of Technology (2004).
[4] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, ArXiv e-prints (2010), 1004.0255.
[5] A. J. Landahl, J. T. Anderson, and P. R. Rice, ArXiv e-prints (2011), 1108.5738.
[6] R. S. Andrist, H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, New Journal of Physics **13**, 083006 (2011).
[7] H. Bombin and M. Martin-Delgado, Phys.Rev.Lett. **97**, 180501 (2006).
[8] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452 (2002).
[9] D. P. Divincenzo and P. Aliferis, Physical Review Letters **98**, 020501 (2007).
[10] A. Kitaev, Annals of Physics **303**, 2 (2003).
[11] J. Edmonds, Canadian Journal of Mathematics **17**, 449 (1965).
[12] S. Bravyi, G. Duclos-Cianci, D. Poulin, and M. Suchara, ArXiv e-prints (2012), 1207.1443.
[13] H. Bombin, G. Duclos-Cianci, and D. Poulin, New Journal of Physics **14**, 073048 (2012).
[14] H. Bombin, ArXiv e-prints (2011), 1107.2707.
[15] H. Bombin, Physical Review A **81**, 32301 (2009).
[16] G. Duclos-Cianci and D. Poulin, ArXiv e-prints (2013), 1302.3638.
[17] J. S. Yedidia, W. T. Freeman, and Y. Weiss, IEEE Transactions on Information Theory **51**, 2282 (2005).
[18] D. Gottesman, Ph.D. thesis, California Institute of Technology (1997).
[19] P. Sarvepalli and R. Raussendorf, Phys. Rev. A **85**, 022317 (2012).
[20] S. Bravyi and J. Haah, ArXiv e-prints (2011).
[21] J. Proakis, *Digital communications*, Communications and signal processing (McGraw-Hill Higher Education, 1995), ISBN 9780070517264.
[22] D. Poulin, Phys. Rev. A **74**, 052333 (2006).
[23] J. R. Wootton and D. Loss, Physical Review Letters **109**, 160503 (2012).
[24] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Phys. Rev. A **86**, 042313 (2012).
[25] E. Knill, Nature (London) **434**, 39 (2005).
[26] M. B. Hastings, Physical Review Letters **101**, 167206 (2008).
[27] D. Poulin, Physical Review Letters **95**, 230504 (2005).
[28] D. Bacon, Phys. Rev. A **73**, 012340 (2006).

Chapitre 8

Article : Subsystem surface codes with three-qubit check operators

Bravyi S., Duclos-Cianci G., Poulin D., Suchara M., *Subsystem surface codes with three-qubit check operators*, Quant. Inf. Comp. Vol 13, No 11&12, pp0963-0985 (2012).

8.1 Contexte

Sergey Bravyi a développé le code de surface à sous-systèmes. Ce code est similaire au code topologique de Kitaev, mais ne nécessite que la mesure d'opérateurs à trois qubits. Dans le but de tester ses performances, Sergey Bravyi et Martin Suchara ont d'abord appliqué une méthode d'appariement pour le décodage, basée sur un modèle de mesure des syndromes en circuit. Toutefois, selon DiVincenzo et Solgun [31], ces mesures à trois qubits pourraient s'effectuer d'un seul coup dans certaines architectures comme les qubits supra-conducteurs. Dans ce cas-ci, Sergey s'attendait à ce qu'une méthode de décodage tenant compte des corrélations ait un avantage. Or, c'est le cas de mon décodeur de renormalisation. Sergey a contacté David qui m'en a fait part. J'ai accepté d'adapter notre méthode à ce cas particulier. Deux difficultés se présentaient. Dans un premier temps, le décodage nécessite la création d'un graphe d'erreurs virtuelles à partir du modèle d'erreurs physiques. Cela donne lieu à un graphe triangulaire, différent des cas traités précédemment. Dans un deuxième temps, il fallait le traduire en modèle de bruit pour les mailles élémentaires cubiques. Ceci résulte en une distribution qui corrèle les erreurs sur les différents qubits d'une même maille

élémentaire. Ces travaux nous ont permis de montrer que le code de surface à sous-systèmes a des seuils de correction d'erreurs et de tolérance aux fautes comparables au code topologique de Kitaev.

8.2 Résumé

La section I introduit l'article et révise le formalisme des codes à sous-systèmes. La section II définit le code de surface à sous-systèmes (CSS). Cette définition n'est pas répétée dans ce chapitre, il est donc important de la lire d'abord. La section III montre que le code topologique de Kitaev et notre nouveau code sont localement équivalents, c.-à-d. qu'ils appartiennent à la même phase topologique. La section IV explique comment encoder des qubits logiques dans une surface planaire plutôt qu'un tore. La section V discute du problème du décodage du CSS. La section VI présente le décodage tolérant aux fautes basé sur une architecture d'extraction du syndrome à l'aide d'un circuit. La section VII présente plutôt le décodage tolérant aux fautes basé sur une extraction directe du syndrome. C'est cette section qui contient l'ensemble de ma contribution. Les sections 8.3 et 8.4 ci-dessous la complètent et doivent être lues après l'article.

8.3 Erreurs

La correction d'erreurs tolérante aux fautes demande de mesurer à répétition le syndrome. Étant donné la structure du stabilisateur et du groupe de jauge, nous choisissons de mesurer en alternance les parties X et Z du syndrome. Chacune de ces parties peut être déduite des résultats de la mesure des opérateurs de jauge triangulaires correspondants. Comme tous les opérateurs de jauge d'un même type commutent, ils peuvent être mesurés simultanément. Plus précisément, tous les opérateurs triangles X sont mesurés en parallèle, puis tout les opérateurs triangles Z sont mesurés en parallèle et ainsi de suite pour toute la durée de l'expérience. Ces mesures sont imparfaites, elles sont affectées par le modèle d'erreurs présenté ci-dessous.

1. Les trois qubits impliqués dans la mesure d'un triangle subissent un canal dépolarisant de paramètre p.
2. Le résultat de mesure est erroné avec une probabilité p.

Le bruit dépolarisant à n qubits applique avec une probabilité p un opérateur de Pauli à n qubits choisi de manière uniformément aléatoire (incluant l'identité). Notons qu'avec ce modèle d'erreurs tous les opérateurs, qui ont un poids variant de 1 à n, ont une probabilité $\mathcal{O}(p)$. De plus, il ne peut être exprimé comme un produit de modèles de bruit indépendants sur chacun des qubits. C'est pourquoi nous disons qu'il corrèle les erreurs sur les différents qubits.

Pour simplifier les simulations, nous décodons les erreurs de type X et Z séparément, ce qui nous permet de ne considérer que les erreurs d'un seul type, X par exemple. Toutefois, comme la mesure des opérateurs triangles fait subir aux qubits un canal dépolarisant qui peut avoir des erreurs X, nous devons toujours considérer l'effet de la mesure des deux types de triangles, même si nous savons que la partie X du syndrome sera triviale. Par conséquent, comme chacun des qubits appartient au support de deux opérateurs de jauge X et de deux opérateurs de jauge Z, ils ne sont jamais au repos (*idle*). Ils sont toujours impliqués dans une mesure et donc la mesure elle-même sera la seule source d'erreurs. De plus, comme nous ne considérons qu'un type d'erreurs, nous pouvons changer la description du bruit en marginalisant sur Z le canal à dépolarisation. Nous appelons le canal résultant « canal dépolarisant X ». De manière similaire au canal dépolarisant, il applique aux qubits, avec probabilité p, un opérateur de Pauli choisi de manière uniformément aléatoire parmi ceux de type X exclusivement.

8.4 Décodage

Dans le but de décoder le code de surface à sous-systèmes, nous le découpons en mailles élémentaires comme le montre la Fig. 8.1. Les qubits présents dans la maille sont ceux qui peuvent causer des syndromes non-triviaux à l'intérieur de celle-ci. C'est pourquoi, par exemple, le qubit qui est sur le site en haut à droite complètement (Fig. 8.1) n'est pas inclus. Une erreur X sur celui-ci ne changerait pas le syndrome de cette maille élémentaire.

Un cycle de mesure du syndrome est défini par une mesure de tous les triangles X et ensuite de tous les triangles Z. Chacune de ces mesures introduit des erreurs. Comme nous l'avons vu, elles sont corrélées. Par contre, dans le but de pouvoir traiter le problème de manière efficace, nous négligeons les corrélations du bruit à l'extérieur de la maille élémentaire. La mesure des divers opérateurs triangles font intervenir un, deux ou trois qubits d'une maille élémentaire donnée. Nous marginalisons sur tout qubit qui est impliqué dans un triangle, mais qui est à l'extérieur de la maille élémentaire considérée. Les Fig. 8.2 et Fig. 8.3 donnent une représentation graphique de tous les canaux dépolarisants X affectant

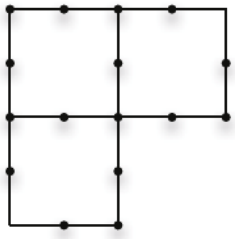


Figure 8.1 Maille élémentaire du CSS et ses 16 qubits.

(a) Sur trois qubits

(b) Sur deux qubits

(c) Sur un qubit

Figure 8.2 Représentation graphique du canal dépolarisant X sur un, deux ou trois qubits.

la maille élémentaire dans un cycle de mesure de syndrome. Notons que chaque qubit intervient dans quatre opérateurs triangles, deux X et deux Z, c'est pourquoi chaque qubit est affecté par quatre canaux, convenablement marginalisés. Le produit de tous ces canaux corrèle les erreurs sur l'ensemble de la maille élémentaire. Ce produit est le modèle d'erreurs à priori que nous utilisons pour décoder.

Nous voulons utiliser le décodeur RG tolérant aux fautes (3D) du code topologique de Kitaev (CTK) pour décoder le code de surface à sous-systèmes. Nous constatons que les qubits sur les sites n'apparaissent pas dans le CTK. Nous devons reformuler le problème pour qu'il soit compatible avec le décodeur RG, c.-à-d. que nous devons transformer les erreurs sur les sites en erreurs sur les arêtes seulement. Pour ce faire, nous utilisons une interprétation en termes de flots de charges. Les erreurs agissant sur les qubits le long des arêtes restent inchangées. Par contre, les erreurs agissant sur les qubits des sites sont plutôt transformées en erreurs à deux qubits le long des arêtes comme l'illustre la Fig. 8.4a. Ce choix est arbitraire, l'essentiel est que l'erreur choisie produise le même syndrome, ce qui est le cas ici. De plus, si la nouvelle erreur devait affecter un qubit à l'extérieur de la maille élémentaire, nous ne considérons que la partie de l'erreur contenue dans la maille, cf. Fig. 8.4b. Il ne reste plus qu'à inclure l'erreur de mesure. Il s'agit d'un simple canal d'inversion de paramètre p. Nous avons donc un syndrome et un modèle d'erreurs à priori complets. Il suffit ensuite de le soumettre au décodeur du CTK tolérant aux fautes adapté aux distributions corrélées sur les mailles élémentaires.

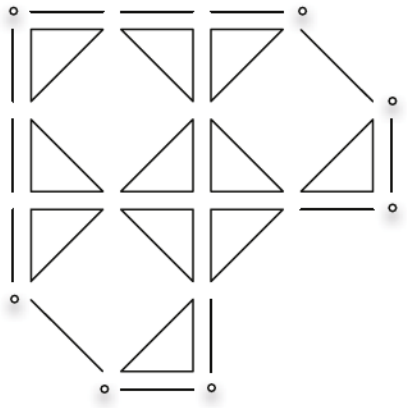
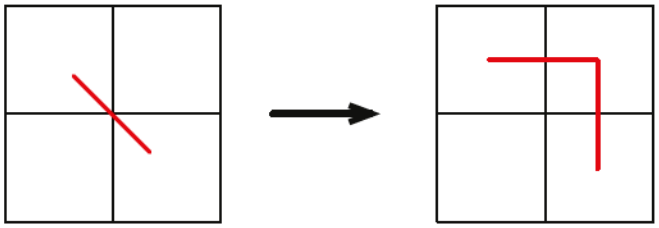
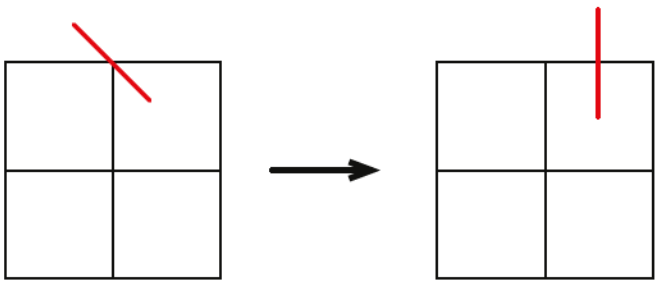


FIGURE 8.3 Canaux dépolarisants X affectant les 16 qubits de la maille élémentaire. Chaque qubit participe à quatre opérateurs triangles. C'est pourquoi chaque qubit est affecté par quatre canaux convenablement marginalisés.



(a) Les erreurs sur les sites sont remplacées par des erreurs à deux qubits sur les arêtes.



(b) Les erreurs affectant des qubits à l'extérieur de la maille élémentaire sont négligées.

FIGURE 8.4 Transformation des erreurs sur les sites en erreurs sur les arêtes.

8.5 Article

Subsystem surface codes with three-qubit check operators

Sergey Bravyi,[1] Guillaume Duclos-Cianci,[2] David Poulin,[2] and Martin Suchara[3]

[1]*IBM Watson Research Center, Yorktown Heights NY 10598 (USA)*
[2]*Département de Physique, Université de Sherbrooke, Sherbrooke, Québec, J1K 2R1 (Canada)*
[3]*Computer Science Division, UC Berkeley, Berkeley CA 94720 (USA)*

We propose a simplified version of the Kitaev's surface code in which error correction requires only three-qubit parity measurements for Pauli operators XXX and ZZZ. The new code belongs to the class of subsystem stabilizer codes. It inherits many favorable properties of the standard surface code such as encoding of multiple logical qubits on a planar lattice with punctured holes, efficient decoding by either minimum-weight matching or renormalization group methods, and high error threshold. The new subsystem surface code (SSC) gives rise to an exactly solvable Hamiltonian with 3-qubit interactions, topologically ordered ground state, and a constant energy gap. We construct a local unitary transformation mapping the SSC Hamiltonian to the one of the ordinary surface code thus showing that the two Hamiltonians belong to the same topological class. We describe error correction protocols for the SSC and determine its error thresholds under several natural error models. In particular, we show that the SSC has error threshold approximately 0.6% for the standard circuit-based error model studied in the literature. We also consider a model in which three-qubit parity operators can be measured directly. We show that the SSC has error threshold approximately 0.97% in this setting.

I. INTRODUCTION

Quantum error correcting codes are vital ingredients in all scalable quantum computing architectures proposed so far. By actively monitoring and correcting errors, the encoded quantum states can be protected from noise up to any desired precision provided that the error rate of elementary quantum operations is below certain constant value known as the error threshold.

Topological codes such as the surface code family [1–3] have received considerable attention lately due to their several attractive features. First, the quantum hardware envisioned in the surface code architecture consists of a 2D array of qubits with controlled nearest-neighbor interactions and a local readout. In principle, it can be implemented using the Josephson junction qubits technology [4]. Surface codes feature an error threshold of at least 1% [5] which is one of the highest thresholds among all studied codes. Secondly, encoded Clifford gates such as the CNOT gate can be implemented efficiently by the code deformation method [6–8] which requires only a mild overhead in space and time. The error rate of encoded gates decreases exponentially with the lattice size [9]. Thirdly, the surface codes can be decoded efficiently using Edmonds's minimum weight matching algorithm [3, 10] or renormalisation group methods [11–13].

Although the surface code is among the best code candidates, a promising direction for improvements has been recently identified by Bombin [14] who proposed topological subsystem codes [15]. A subsystem code [16, 17] can be viewed as a regular stabilizer code in which one or several logical qubits do not encode any information. The presence of unused logical qubits, known as *gauge qubits*, simplifies eigenvalue measurements of multi-qubit stabilizers—such as the plaquette and star operators of the surface code—which are required for error correction. Consider as an example the simplest 4-qubit code with two stabilizers $S^X = X^{\otimes 4}$ and $S^Z = Z^{\otimes 4}$. It encodes two qubits with logical Pauli operators $\overline{X}_L = X_1X_2$, $\overline{Z}_L = Z_1Z_3$ and $\overline{X}_G = X_1X_3$, $\overline{Z}_G = Z_1Z_2$. If only the first logical qubit is used to encode information, the syndrome (eigenvalue) of S^X can be determined indirectly by measuring eigenvalues of the unused logical operators $\overline{X}_G$ and $S^X\overline{X}_G = X_2X_4$. Multiplying the measured eigenvalues together yields the desired eigenvalue of S^X. The syndrome of S^Z is determined similarly by measuring eigenvalues of the unused logical operators $\overline{Z}_G$ and $S^Z\overline{Z}_G = Z_3Z_4$ followed by multiplication of the outcomes. Hence the full syndrome extraction requires only two-qubit parity measurements and simple classical post-processing. The unused logical operators that need to be measured in order to extract the syndrome of all stabilizers are usually referred to as *gauge generators*, see [17] for the general theory of subsystem codes.

A simplified syndrome readout offered by subsystem codes has its own costs. In any practical settings, eigenvalues of individual gauge generators can only be measured with a finite accuracy. As one multiplies together measured eigenvalues, errors tend to accumulate rendering the inferred syndrome bit unreliable. This strongly limits the class of candidate subsystem codes for the topological fault-tolerant architecture. First, a suitable code must have local gauge generators, ideally, 2- or 3-qubit Pauli operators acting on nearest-neighbor qubits. This ensures that the syndrome readout requires only local measurements. To avoid accumulation of measurement errors, a suitable code must also have low-weight stabilizers. More precisely, each stabilizer must be composed of only a few gauge generators. The latter requirement leaves out many interesting families of codes, such as the 2D Bacon-Shor codes [16] and random 2D subsystem codes discovered in [18]. In contrast, subsystem color codes found in [14] have 2-qubit gauge generators while stabilizers act on either 6 or 18 qubits. These codes

were shown to have a constant error threshold of at least 2% under depolarizing noise assuming noiseless syndrome readout [19, 20]. Unfortunately, subsystem color codes do not inherit favorable properties of the surface code such as encoding of multiple logical qubits [14] on a planar lattice required for the code deformation method.

In the present paper we propose a subsystem version of the standard surface code on the regular square lattice. Each plaquette of the lattice carries one gauge qubit and a pair of weight-6 stabilizers of type $X^{\otimes 6}$ and $Z^{\otimes 6}$, see Fig. 1. The code has 3-qubit gauge generators of type XXX and ZZZ which makes it suitable for architectures where direct 3-qubit parity measurements in the X- and Z-basis are available. A promising proposal for implementing 3-qubit parity measurements in Josephson junction qubits has been recently made by DiVincenzo and Solgun [21]. By analogy with the surface code, the new subsystem surface code (SSC) [illegible] logical qubits on a planar lattice v [illegible] We describe error correction proto [illegible] determine its error thresholds und [illegible] ror models. First, we study the so c [illegible] model where each qubit is subject t [illegible] and phase-flip errors with rate p, [illegible] surements are noiseless. By relati [illegible] correction to the phase transition [illegible] Ising model on the honeycomb latt [illegible] threshold error rate is $p_0 \approx 7\%$. Se [illegible] circuit-based error model where the [illegible] simulated by noisy quantum gates, [illegible] ments, and ancilla preparations. Ea [illegible] with a probability p, see Section V [illegible] Carlo simulation suggests that the [illegible] old $p_c \approx 0.6\%$ for the circuit-based [illegible] we consider a model in which 3-qub [illegible] sured directly. We show that the SS [illegible] approximately 0.97% for this direct [illegible] model.

The new code also gives rise to [illegible] Hamiltonian with 3-qubit interactio [illegible] logically ordered ground state and [illegible] non-interacting abelian anyons. Th [illegible] the model stems from a peculiar [illegible] ture of the model's Hamiltonian. [illegible] of all relevant 3-qubit interactions ca [illegible] small clusters such that interaction [illegible] ters pairwise commute. In contra [illegible] cently by Aharonov and Eldar [22] that topological order cannot be realized by 3-local Hamiltonians in which *all* interactions pairwise commute. We also construct a local unitary transformation U that maps the 6-qubit stabilizers of the SSC to the plaquette and star operators of the Kitaev's surface code on the square lattice. The gauge generators XXX and ZZZ are mapped to single-qubit X and Z operators. Loosely speaking, the map U decouples gauge qubits from the surface code qubits.

The rest of the paper is organized as follows. Sections II,III introduce a subsystem version of the toric code with 3-qubit gauge generators, the corresponding exactly solvable Hamiltonian and discuss its connection to the ordinary toric code. Extension to the planar geometry is given in Section IV that defines subsystem surface codes. The concept of a virtual lattice which is crucial for understanding our error correction protocols is introduced in Section V. This section also discusses error correction for the idealized settings when syndrome readout is noiseless. Error correction protocols for the circuit-based syndrome readout and numerical simulations are presented in Section VI. Finally, Section VII focuses on a model in which direct 3-qubit parity measurement are available.

II. A SUBSYSTEM TORIC CODE

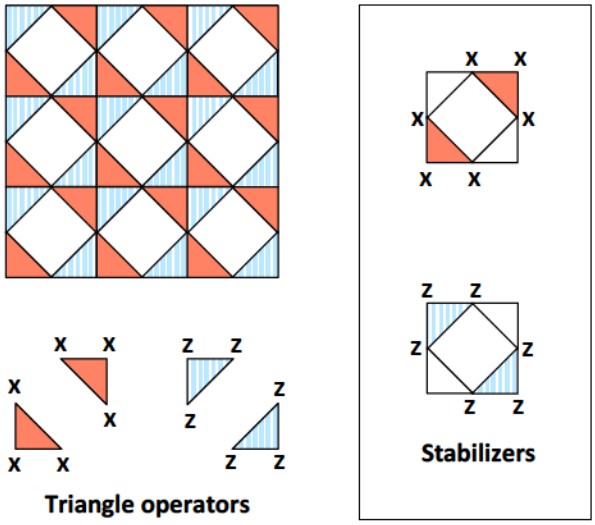


FIG. 1: Subsystem toric code. Qubits live at vertices and centers of edges of the regular square lattice. Opposite sides of the lattice are identified. *Left:* Four types of triangles and the corresponding triangle operators $G(T)$. Triangle operators that belong to different plaquettes pairwise commute. *Right:* Stabilizer operators S_p^X (top) and S_p^Z (bottom). Stabilizers are analogues of the plaquette and star operators of the standard toric code. Triangle operators commute with stabilizers. Eigenvalue of any stabilizer can be determined by measuring eigenvalues of individual triangle operators. For a lattice of size $L \times L$ the code has parameters $[[3L^2, 2, L]]$. This should be compared with the standard surface code which has parameters $[[2L^2, 2, L]]$.

operators S_p^X and S_p^Z as shown on Fig. 1 (right). One can easily check that S_p^X and S_q^Z commute with each other for all p and q. Let $\mathcal{S}$ be the abelian group generated by $2L^2$ operators S_p^X and S_p^Z. It defines a quantum stabilizer code with a codespace $\mathcal{C}$ spanned by n-qubit states ψ invariant under $\mathcal{S}$, that is, $\psi \in \mathcal{C}$ iff $S_p^X \cdot \psi = \psi$ and $S_p^Z \cdot \psi = \psi$ for all p. A simple algebra shows that $\prod_p S_p^X = I$ and $\prod_p S_p^Z = I$, where the product is taken over all plaquettes of the lattice. Furthermore, since each qubit belongs to exactly two stabilizers S_p^X and two stabilizers S_p^Z, these are the only dependencies among the generators of $\mathcal{S}$. This shows that $\mathcal{S}$ has $s = 2(L^2 - 1)$ independent generators. The standard stabilizer formalism [23] then implies that $\mathcal{S}$ is a stabilizer code enc $k' = n - s = L^2 + 2$ qubits, that is, $\dim(\mathcal{C}) = 2^{k'}$.

We shall now divide the k' encoded qubits into g gauge qubits and $k = 2$ logical qubits. Let u b vertex of the lattice and f, g be a pair of orthogonal incident to u. The triple (u, f, g) will be referred a *triangle*. Note that the lattice has four non-equi types of triangles, see Fig. 1. We shall say that a tr $T = (u, f, g)$ is north-west (NW) if u is at the nort corner of the plaquette formed by f and g. Similar defines north-east (NE), south-west (SW), and sout (SE) triangles. Define *triangle operators*

$$G(T) = \begin{cases} X_u X_f X_g & \text{if } T \text{ is SW or NE triangle,} \\ Z_u Z_f Z_g & \text{if } T \text{ is SE or NW triangle,} \end{cases}$$

see Fig. 1. Here subscripts indicate qubits acted up the Pauli operators X and Z. Note that triangle ators that belong to different plaquettes commute each other.

Any stabilizer can be expressed as a product c triangle operators using identities

$$\begin{aligned} S_p^X &= G(T_p^{SW}) G(T_p^{NE}), \\ S_p^Z &= G(T_p^{SE}) G(T_p^{NW}), \end{aligned}$$

see Fig. 1. Here T_p^{NW}, T_p^{NE}, T_p^{SW}, and T_p^{SE} are tri of type NW, NE, SW, SE respectively that belon plaquette p.

Let us now show that triangle operators commute with all stabilizers, thus being logical operators of the code $\mathcal{S}$. Consider any stabilizer, say, S_p^X and any triangle operator $G(T)$ of Z-type. If T does not belong to the plaquette p then S_p^X commutes with $G(T)$ because triangle operators from different plaquettes always commute. If T belongs to p then $G(T)$ anti-commutes with both X-type triangles forming S_p^X, that is, $G(T)$ commutes with S_p^X. A similar argument shows that Z-type stabilizers commute with X-type triangle operators.

The above observations show that we can choose $g = L^2$ pairs of logical operators for the code $\mathcal{S}$ as

$$\overline{X}_p = G(T_p^{SW}) \quad \text{and} \quad \overline{Z}_p = G(T_p^{SE}), \tag{3}$$

where p runs over all plaquettes of the lattice. We shall treat encoded qubits defined by $\overline{X}_p$ and $\overline{Z}_p$ as gauge qubits encoding no useful information because the corresponding logical operators have very small weight. Hence each plaquette of the lattice carries one gauge qubit. As we will show in Sec. III, it is possible to completely disentangle these gauge qubits from the code with a depth 4 local quantum circuit, leaving behind the usual toric code on $2L^2$ qubits and L^2 ancillary qubits that are decoupled from the code.

Recall that the code $\mathcal{S}$ has $k' = L^2 + 2$ encoded qubits. This leaves $k = k' - g = 2$ logical qubits which have not been identified yet. Let Γ and Λ be the set of all qubits lying on some fixed horizontal and some fixed vertical line of the lattice respectively, see Fig. 2. Note that $|\Gamma| = |\Lambda| = 2L$. Define

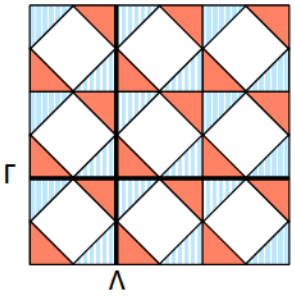


FIG. 2: Non-contractible loops on the toric code lattice. Each loop contains $2L$ qubits.

Any triangle operator commutes with $\overline{X}_j$ and $\overline{Z}_j$ because triangles share 0 or 2 qubits with Γ and Λ. In addition, since Γ and Λ are even sets of qubits overlapping on exactly one qubit, one has commutation rules $\overline{X}_i \overline{Z}_j = (-1)^{\delta_{i,j}} \overline{Z}_j \overline{X}_i$. Hence one can view $\overline{X}_1, \overline{Z}_1$ and $\overline{X}_2, \overline{Z}_2$ as logical X and Z operators on the two remaining logical qubits of the code $\mathcal{S}$.

The minimum distance d of a subsystem code is defined as the minimum weight of a Pauli error E that commutes with all stabilizers and implements a non-trivial transformation on the logical qubits, see [17]. Let us show that the subsystem toric code has distance $d = L$. Indeed, any error E as above must anti-commute with at least one of the logical operators $\overline{X}_1, \overline{Z}_1, \overline{X}_2, \overline{Z}_2$. Assume wlog that E anti-commutes with $\overline{Z}_1$. Let Λ' be any vertical line on the lattice (a horizontal translation of Λ) and $\overline{Z}'_1 = \prod_{j \in \Lambda'} Z_j$. One can easily check that $\overline{Z}_1 \overline{Z}'_1$ coincides with the product of stabilizers S_p^Z over all plaquettes p lying between Λ and Λ'. Since E commutes with all stabilizers, we conclude that E anti-commutes with

$\overline{Z}'_1$. But this means that E must act non-trivially on at least one qubit of Λ'. Since there are L non-overlapping choices of the line Λ', we conclude that E must have weight at least L. One can also easily check that translating Λ by the half of the lattice period gives a logical operator of weight L equivalent to $\overline{Z}_1$ (modulo gauge operators). Hence the code has distance $d = L$.

The first step in any error correction protocol based on stabilizer codes is the *syndrome readout*, that is, a non-destructive eigenvalue measurement of every stabilizer operator. To measure the 6-qubit stabilizers S_p^Z and S_p^X we shall take advantage of the gauge qubits and the identity Eq. (2). The simplest syndrome readout protocol consists of two steps: *Step 1.* Measure the eigenvalue of every X-type triangle operator $G(T)$ and record the outcome $\lambda(T) = \pm 1$. *Step 2.* Measure the eigenvalue of every Z-type triangle operator $G(T)$ and record the outcome $\lambda(T) = \pm 1$. Since any triangle operator commutes with stabilizers, the eigenvalue of any stabilizer remains unchanged throughout the execution of this protocol. Hence the eigenvalues of stabilizers S_p^Z and S_p^X are given by $\lambda(S_p^Z) = \lambda(T_p^{SE})\lambda(T_p^{NW})$ and $\lambda(S_p^X) = \lambda(T_p^{SW})\lambda(T_p^{NE})$, see Eq. (2). In practice it may be advantageous to use 'interleaved' protocols in which Steps 1,2 defined above are implemented in parallel, see Section VI for more details.

III. TOPOLOGICAL QUANTUM ORDER

Consider a Hamiltonian

$$H = -\sum_T G(T), \tag{6}$$

where the sum is over all triangles of the lattice. Recall that $G(T)$ are the 3-qubit triangle operators defined in Eq. (1). In this section we compute the entire eigenvalue spectrum of H and show that on the torus H has a four-fold degenerate ground state separated from excited states by a constant energy gap. Moreover, we shall construct a unitary locality preserving transformation U such that $UHU^\dagger$ can be regarded as the standard toric code Hamiltonian on the square lattice (with some irrelevant ancillary qubits). Thus the model defined in Eq. (6) exhibits topological quantum order.

Let us first compute eigenvalues of H. Since the stabilizers S_p^X, S_p^Z commute with every term in H, we can assume that any eigenvector ψ of H is also an eigenvector of any stabilizer, that is, $S_p^X\psi = x_p\psi$ and $S_p^Z\psi = z_p\psi$ for some syndromes $x_p, z_p = \pm 1$. Using identities Eq. (2,3) one gets

$$H\psi = -\sum_p (1+x_p)\overline{X}_p\psi + (1+z_p)\overline{Z}_p\psi$$

where $\overline{X}_p, \overline{Z}_p$ are the logical operators on the gauge qubit located at the plaquette p. Hence the restriction of H onto the sector with fixed syndromes x_p, z_p describes L^2 non-interacting gauge qubits.

Let $\epsilon_0(x_p, z_p)$ and $\epsilon_1(x_p, z_p)$ be the smallest and the largest eigenvalues of a gauge qubit p for a fixed syndromes x_p, z_p. A simple algebra shows that

x_p	z_p	$\epsilon_0(x_p, z_p)$	$\epsilon_1(x_p, z_p)$
1	1	$-2\sqrt{2}$	$2\sqrt{2}$
1	-1	-2	2
-1	1	-2	2
-1	-1	0	0

To minimize the overall energy one has to choose $x_p = z_p = 1$ for all p. This shows that ground states of H belong to the trivial syndrome sector and the ground state energy is $E_0 = -2\sqrt{2}L^2$. The ground state is four-fold degenerate since the code has two logical qubits.

Excitations of H fall into two categories. First, there are gauge excitations that are confined to the trivial syndrome subspace $x_p = z_p = 1$. The energy cost of a single gauge excitation is $\Delta_g = \epsilon_1(1,1) - \epsilon_0(1,1) = 4\sqrt{2}$. A gauge excitation on a plaquette p can be created locally by a proper combination of operators $\overline{X}_p$ and $\overline{Z}_p$. Hence gauge excitations do not carry any topological charge. Secondly, there are syndrome excitations that flip syndrome bits x_p and z_p. The energy cost of a single syndrome excitation is $\Delta_s = \epsilon_0(1,-1) - \epsilon_0(1,1) = 2(\sqrt{2}-1)$. It corresponds to flipping x_p (or z_p) on any plaquette p. A single syndrome excitation however cannot be created locally due to the constraints $\prod_p x_p = \prod_p z_p = 1$, see the previous section. It means that syndrome excitations can only be created in pairs. Each pair costs energy $2\Delta_s$.

We can now show that the Hamiltonian of Eq. (6) is locally equivalent to Kitaev's toric code. Consider a quantum circuit U shown on Fig. 3. It consists of four rounds of CNOT gates, $U = U^{(4)}U^{(3)}U^{(2)}U^{(1)}$, where $U^{(j)}$ is a tensor product of L^2 disjoint CNOT gates labeled by j on Fig. 3. Note that U is a locality preserving transformation, that is, the Heisenberg evolution of any observable $O \to UOU^\dagger$ can only enlarge the support of O by a few units of length. Such transformations do not change any topological features of the model [24]. A simple algebra shows that the transformed stabilizers $US_p^XU^\dagger \equiv A_p$ and $US_p^ZU^\dagger \equiv B_p$ coincide with the star and plaquette operators of the Kitaev's toric code on a tilted square lattice, see Fig. 3. Furthermore, the transformed gauge generators $U\overline{X}_pU^\dagger \equiv J_p^X$ and $U\overline{Z}_pU^\dagger \equiv J_p^Z$ become one-qubit Pauli operators X and Z respectively acting on the qubit located at the bottom edge of p, see Fig. 3. Using identities Eqs. (2,3) we arrive at

$$H' \equiv UHU^\dagger = -\sum_p J_p^X + J_p^X A_p + J_p^Z + J_p^Z B_p.$$

The same arguments as above show that ground states of H' are defined by equations $A_p\psi_0 = B_p\psi_0 = \psi_0$ and $(J_p^X + J_p^Z)\psi_0 = \sqrt{2}\psi_0$ for all plaquettes p. Thus any ground state of H' must have a form $\psi_0 = \psi_{top} \otimes \psi_{anc}$,

where ψ_{top} is a ground state of the Kitaev's toric code on the tilted square lattice while ψ_{anc} is a tensor product of one-qubit ancillary states located on horizontal edges of the original lattice. Such ancillary unentangled states clearly have no effect on topological features of the model. We conclude that that the Hamiltonian Eq. (6) is in the same topological phase as the toric code model. The exact solvability of the model clearly extends to a more general Hamiltonian $H = -\sum_T g_T G(T)$, where g_T are arbitrary coefficients.

Consider now a modified Hamiltonian

$$H'' = -\sum_p J_p^X + J_p^Z + A_p + B_p.$$

Note that H' and H'' have the same ground subspace and H'' coincides with the ordinary toric code Hamiltonian [1] if one ignores the ancillary qubits. For any state ϕ orthogonal to the common ground subspace of H', H'' and for any parameter $0 \le t \le 1$ one has

$$\langle\phi|(1-t)H' + tH''|\phi\rangle \ge (1-t)\Delta' + t\Delta''$$

where $\Delta' = 4(\sqrt{2}-1)$ and $\Delta'' = 2$ are the energy gaps of H' and H'' respectively. It follows that $(1-t)H' + tH''$ has energy gap at least $4(\sqrt{2}-1)$ for all $0 \le t \le 1$. Hence we

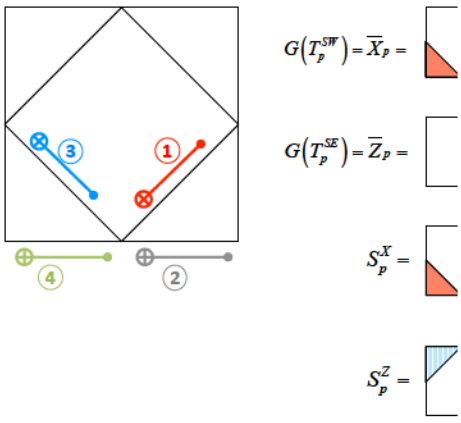


FIG. 3: Decoupling circuit. Applying fou gates as shown on the left in a translati transforms the gauge operators and stab the right. The stabilizer generators become those of Kitaev's toric code on a tilted square lattice, while the extra gauge generators are mapped to single-qubit Pauli operators acting on ancillary qubits (horizontal edges). Thus, the circuit has the effect of decoupling the gauge operators from the toric code.

We note that in general the Hamiltonian obtained by the sum of the gauge operators of a topological subsystem code does not necessarily produce topological order. The peculiar feature of the present model is that the canonical logical Pauli operators on the gauge qubits $\overline{X}_p$ and $\overline{Z}_p$ are *local*. In contrast, it was shown in [25] that the subsystem color code can be obtained from multiple copies of the ordinary color code by gauging out both local and non-local logical operators. In particular, some of the logical operators that are gauged out carry topological charge. The present analysis does not apply to such models.

IV. SUBSYSTEM SURFACE CODES

We can now describe a subsystem version of the simplest surface code on a planar lattice with two rough and smooth boundaries that encodes one logical qubit [2]. Now the lattice has open boundary conditions. A lattice of size $L \times L$ has $(L+1)^2$ vertices, $2L(L+1)$ edges, and L^2 plaquettes. As before, code qubits are placed at vertices and centers of edges, so the total number of code qubits is $n = (L+1)^2 + 2L(L+1) = 3L^2 + 4L + 1$.

For every edge e lying on the boundary of the lattice

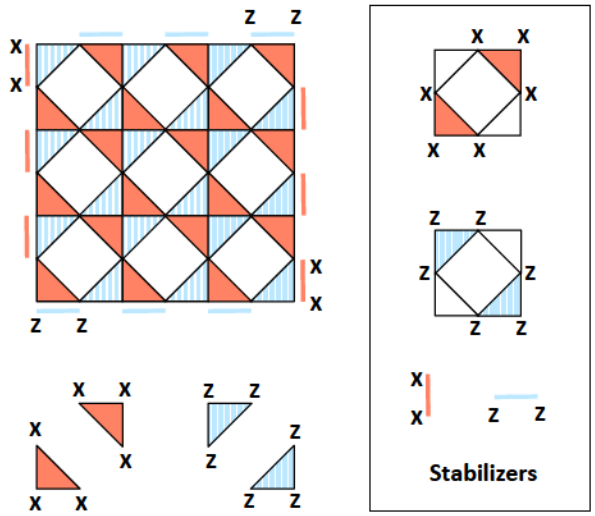


FIG. 4: Subsystem version of the surface code.

Since the additional weight-2 stabilizers S_e lying on the boundary commute with all triangle operators $G(T)$, we can use Eq. (3) to define $g = L^2$ gauge qubits associated with plaquettes of the lattice. Logical operators on the remaining $k = k' - g = 1$ logical qubit can be chosen as $\overline{X}_1$ and $\overline{Z}_1$, see Eq. (4), that is, a horizontal line of X's and a vertical line of Z's. Note that a vertical line of X's and a horizontal line of Z's are no longer logical operators

because they anti-commute with some of the boundary stabilizers S_e. The same arguments as above imply that the code has minimum distance $d = L$. An extension to a planar lattice with punctured holes encoding multiple logical qubits is sketched on Fig. 5.

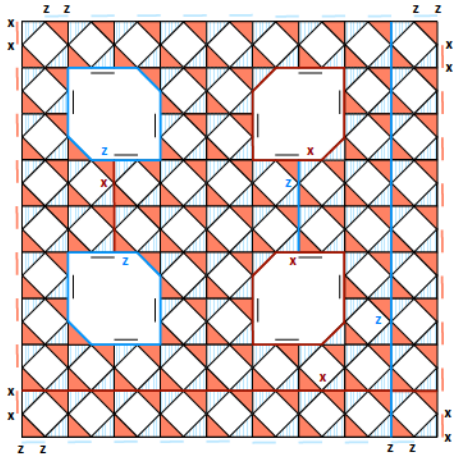


FIG. 5: Subsystem version of the surface code with two punctured holes. Each pair of holes encodes one logical qubit. The external boundary of the lattice generates the third logical qubit. Thick red and blue lines show the logical operators.

V. ERROR CORRECTION WITH NOISELESS SYNDROMES

In this section we propose an error correction protocol for the idealized setting where the syndrome readout is noiseless. For simplicity, we shall first focus on the subsystem toric code defined in Section II. It suffices to construct a protocol for correcting errors of X-type (bit flips). Due to the code symmetry, the same protocol can be applied to Z-type errors. Let E be an unknown Pauli error of X-type. We will say that E creates a *defect* at a plaquette p iff E anti-commutes with the stabilizer S_p^Z. The syndrome measurement reveals a configuration of defects created by E. The key observation is that any single-qubit X-error creates exactly two defects. Indeed, an X error on a vertical or horizontal edge e creates defects at the two plaquettes adjacent to e. An X error at any vertex u creates defects at the two plaquettes lying in the north-west and the south-east quadrants of u. The relationship between single-qubit X-errors and the corresponding pairs of defects can be captured by introducing a *virtual lattice* Λ that consists of virtual vertices and virtual edges. A virtual vertex p represents a stabilizer S_p^Z (plaquette p of the original lattice), while a virtual edge represents a pair of defects that can be created by a single-qubit X error. One can easily check that Λ is the regular triangular lattice, see Fig. 6.

Furthermore, for each virtual edge $e = (p, q)$ there is only one X-error creating a pair of defects at p and q. For any virtual vertex p let $\delta(p)$ be the set of virtual edges incident to p. Then

$$S_p^Z = \prod_{e \in \delta(p)} Z_e,$$

that is, Z-type stabilizers can be regarded as star operators of the standard toric code defined on the triangular lattice. Note that a closed loop on the virtual lattice enclosing any triangular face is an X-type triangle operator $G(T)$, while non-contractible closed loops correspond to logical operators.

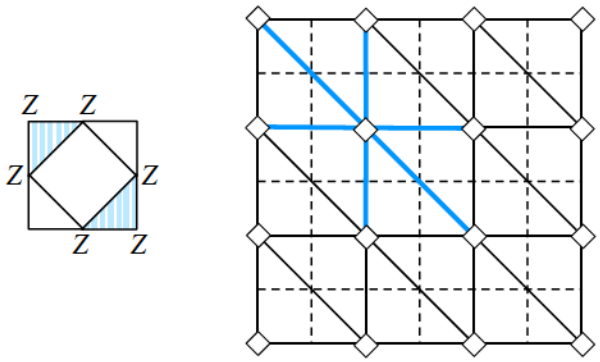


FIG. 6: The virtual lattice Λ (solid lines) describing correction of X-type errors for the subsystem toric code of Fig. 1. Opposite sides of the lattice must be identified. The original square lattice is shown by dashed lines. Each virtual edge represents a code qubit. Stabilizers of Z-type S_p^Z correspond to stars on the virtual lattice (solid blue lines). Triangle operators of X-type correspond to triangular faces of the virtual lattice (not shown). Error correction amounts to finding the minimum weight matching of defects on the virtual lattice. The virtual lattice describing correction of Z-type errors is obtained from Λ by the 90° rotation.

Assuming that errors on different qubits are independent and have the same rate p, the most likely error E^* consistent with the observed syndrome coincides with the minimum weight matching of defects on the virtual lattice. The latter can be found efficiently using the Edmonds's algorithm, see [3] for details. Choosing E^* as a correction operator always returns the system back to the codespace $\mathcal{C}$. The overall evolution of the system is described by an operator EE^* which has trivial syndrome and thus can be viewed as a linear combination of closed loops on the virtual lattice with $\mathbb{Z}_2$ coefficients. Error correction is successful iff EE^* acts non-trivially only on the gauge qubits, that is, EE^* is a product of X-type triangle operators $G(T)$. Equivalently, EE^* must represent the trivial cycle in the homology group $H_1(\Lambda, \mathbb{Z}_2)$.

As was argued in [3, 11, 14], the optimal error correction strategy amounts to finding the most likely *equivalence class* of errors consistent with the observed syndrome rather than the most likely error. More specifically, let $\mathcal{G}$ be the *gauge group* generated by the triangle operators $G(T)$ of X-type. Since logical operators acting on gauge qubits are irrelevant, all errors in the coset

$E \cdot \mathcal{G}$ must be considered equivalent. Note that there are only four cosets of $\mathcal{G}$ consistent with the observed syndrome, namely, $E \cdot \mathcal{G}$, $E\overline{X}_1 \cdot \mathcal{G}$, $E\overline{X}_2 \cdot \mathcal{G}$, and $E\overline{X}_1\overline{X}_2 \cdot \mathcal{G}$. As was shown in [3, 14], the probability of each coset can be expressed as the partition function of the random bond $\pm J$ Ising model. In our case Ising spins reside on triangular faces of the virtual lattice, anti-ferromagnetic bonds correspond to virtual edges that belong to E, and the inverse temperature β is determined by the Nishimori condition $e^{-2\beta J} = \frac{p}{1-p}$. The threshold error rate p_c for the optimal decoding coincides with the density of anti-ferromagnetic bonds at the phase transition point [3]. The latter has been recently computed by Queiroz [26] who found $p_c \approx 7\%$. The analogous threshold error rate for the standard toric code is known to be approximately 11%, see [3].

One can similarly construct the virtual lattice for the subsystem surface code, see Fig. 7. The only difference is that now defects can be matched either to each other, or to one of the boundaries.

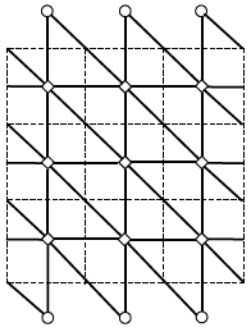


FIG. 7: The virtual lattice Λ (solid lines) describing correction of X-type errors for the subsystem surface code of Fig. 4. Diamonds and open circles represent stabilizers Z-type stabilizers S_p^Z and S_e respectively. The virtual lattice describing correction of Z-type errors is obtained from Λ by the 90° rotation.

VI. ERROR CORRECTION FOR THE CIRCUIT-BASED ERROR MODEL

Let us now consider more realistic settings when the syndrome information itself may contain errors. We assume that the library of elementary operations supported by the quantum hardware includes CNOT gates between nearest-neighbor qubits, single-qubit measurements in X- or Z-basis, and preparation of single-qubit ancillary states $|0\rangle$ and $|+\rangle$. Our error correction protocol will be defined as a sequence of *rounds*, where at each round any qubit can participate in one elementary operation. We assume that each elementary operation is noisy, so it can fail with a probability p that we call an *error rate*. More precisely, our error model, borrowed from [8], is defined as follows.

- A noisy X or Z measurement is the ideal measurement in which the outcome is flipped with probability p.
- A noisy $|0\rangle$ or $|+\rangle$ ancilla preparation returns the correct state with probability $1-p$ and the orthogonal state with probability p.
- A noisy CNOT gate is the ideal CNOT followed by an error $(1-p)Id + p\mathcal{D}$, where Id is the identity map and $\mathcal{D}$ is the fully depolarizing two-qubit map applying one of 16 two-qubit Pauli operators with probability 1/16 each.

We assume that once a qubit has been measured, its state is unknown. To use such a qubit again, it must be explicitly initialized using the noisy preparation defined above. We do not need to define memory errors because no qubit will be idle at any round of our protocol.

In order to measure eigenvalue of individual triangle operators $Z_iZ_jZ_k$ and $X_pX_qX_r$ we shall use quantum circuits shown on Fig. 8. Measuring a single triangle operator requires one ancillary qubit and five rounds. Similar circuits with one extra CNOT gate were used in fault-tolerant protocols based on the standard surface code [3, 8], where one has to measure four-qubit plaquette and star operators $Z^{\otimes 4}$ and $X^{\otimes 4}$.

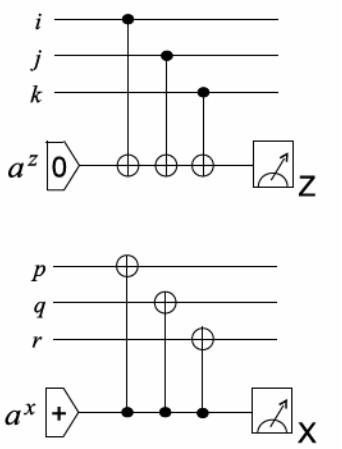


FIG. 8: Quantum circuits for measuring the eigenvalue of triangle operators $Z_iZ_jZ_k$ (top) and $X_pX_qX_r$ (bottom). The circuits use one ancillary qubit. A single Z (X) error on the ancilla a^z (a^x) can propagate via CNOTs to at most one Z (X) error on code qubits modulo gauge operators. A single X (Z) error on the ancilla a^z (a^x) results in a faulty measurement outcome.

We begin by highlighting strengths and weaknesses of the subsystem and the standard surface code. The key advantage of the SSC is a limited propagation of errors by the CNOT gates. Consider, for example, a single Z error on the ancillary qubit a^z in the circuit measuring the triangle operator $Z_iZ_jZ_k$, see Fig. 8 (top). Depending

on the round at which the error has occurred, it propagates to one of the errors Z_i, Z_iZ_j, $Z_iZ_jZ_k$ on the code qubits. Multiplying the last two errors by the triangle operator $Z_iZ_jZ_k$ leaves only single-qubit errors Z_i, Z_k, and the identity error. It shows that a single Z error on the ancilla a^z can lead to at most one Z error on the code qubits, modulo gauge operators. An X error on a^z cannot propagate to code qubits, so its only effect is flipping the measurement outcome which results in a faulty syndrome bit. Since each stabilizer is composed of two disjoint triangle operators, see Fig. 4, each syndrome bit effectively accumulates errors from ten rounds. For comparison, the standard surface code requires six rounds to measure a single syndrome bit, however a single error on the ancilla can feed back to two errors on code qubits (such double errors were referred to as 'horizontal hooks' in [3]). This shows that neither of the
an obvious advantage compared with th

Let us now discuss our syndrome re
more details. Since individual syndrom
be trusted, we shall repeat syndrome r
times for some $T \gg 1$. We choose $T =$
cal simulations. Error correction is deer
the accumulated error E on the code qu
from T noisy syndrome measurements
based on the full observed syndrome hist
syndrome readout which we assume to
practice the final readout involves meas
qubit in $|0\rangle$ or $|+\rangle$ basis. Outcomes of su
determine the syndrome of Z-type or X
respectively. We can assume that single
ments are noiseless by absorbing measure
memory errors that occurred one round

Repeating the circuits shown on Fig
times would naively require $5L$ rounds.
the required number of rounds to $4L$ by
ancillary qubits for each triangle. One o
the ancilla a^z or a^x shown on Fig. 8. Th
second ancilla is to enable offline prepara
states which can be performed in the sa
measurement of the main ancilla. To sir
we will only show one ancilla per trian
that this ancilla is initialized in the $|0\rangle$
the end of each measurement round (wi
noise).

The readout circuit will be chosen suc
triangle alternates between three gate
measurement round in a cyclic fashion.
sented by a local readout schedule

$$\cdots MG_1G_2G_3MG_1G_2G_3M$$

where M is either X-type or Z-type measurement on the ancilla while G_1, G_2, G_3 are CNOT gates coupling the ancilla and the code qubits (the ancilla is control for X-type triangles and target for Z-type triangles). Time flows from the left to the right. Note that each triangle has 24 different choices of its local schedule. Indeed, there are 6 choices of the order in which the ancilla is coupled to the code qubits and 4 choices of the round at which the triangle is measured. Local schedules chosen at different triangles must be consistent with each other, such that at any round any qubit participates in at most one operation.

We shall focus on schedules which are periodic both in space and time. Hence the entire readout circuit is completely specified by local schedules inside a 3D elementary cell which consists of four rounds labeled as $0, 1, 2, 3$ and four triangles of type NE, SE, SW, NW located at some fixed plaquette p. We begin by observing that a consistent schedule cannot have a round at which every triangle applies a CNOT. Indeed, this would define a matching between code qubits and triangles. However, the lattice has $4L^2$ triangles and only $3L^2$ code qubits. This observation shows that at every round all triangles

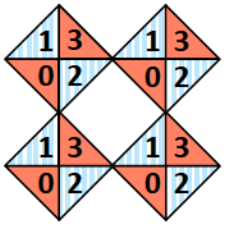


FIG. 9: The numbers indicate rounds (modulo four) at which the ancillas assigned to each triangle have to be measured.

It remains to schedule CNOT gates. We will say that a schedule is *correct* iff for each triangle one can move all gates forward in time towards the next measurement. Here moving a gate is allowed as long as it commutes with other gates. A correct schedule faithfully simulates the simple syndrome extraction routine described in Section II since after moving all gates towards the next measurement all X-type stabilizers are measured at rounds $3, 0$, while all Z-type stabilizers are measured at rounds $1, 2$. We shall look for a schedule which is correct and invariant under the exchange of X and Z (modulo lattice symmetries and time translations). The latter requirement is rather natural since our error model does not have a bias towards X or Z errors.

To derive sufficient conditions fc
introduce some terminology. Consid
its local schedule, see Eq. (7). We sh
G_1 and G_3 as the *first gate* and the *la*
triangle. If some pair of triangles T
at two subsequent rounds j and j
will say that T is the *leading triangl*
triangle. If T and T' are measured t
choice of the leading and the tailing

Lemma 1. *A schedule of CNOTs is*
type and any Z-type triangle at leas
is true:

- *The two triangles are disjoint,*
- *The two triangles are measure*
- *The last gate of the leading tri*
  the first gate of the tailing tri

Proof. Suppose T^x and T^z are X-ty
gles respectively. If T^x and T^z are r
apart, their combined local schedule
by a diagram

$\cdots$	M^x	G_1^x	G_2^x	G_3^x	M^x	(
$\cdots$	G_2^z	G_3^z	M^z	G_1^z	G_2^z	(

The gates G_1^x and G_3^z must be disjoint. Similarly, the gates G_3^x and G_1^z must be disjoint. Hence we can deform the diagram by moving G_1^x, G_1^z one round forward and moving G_3^x, G_3^z one round backward obtaining an equivalent circuit:

$\cdots$	M^x	$G_1^xG_2^xG_3^x$	M^x	$G_1^xG_2^xG_3^x$ $\cdots$
$\cdots$	$G_1^zG_2^zG_3^z$	M^z	$G_1^zG_2^zG_3^z$	M^z $\cdots$

We can further deform the circuit by moving each measurement backwards towards the next gate.

Suppose now that T^x and T^z are measured in two subsequent rounds such that T^x is the leading and T^z is the tailing (the opposite case is completely analogous). Then their combined schedules can be represented by a diagram

$\cdots$	M^x	G_1^x	G_2^x	G_3^x	M^x	G_1^x	$\cdots$
$\cdots$	G_3^z	M^z	G_1^z	G_2^z	G_3^z	M^z	$\cdots$

By assumption, the gates G_3^x and G_1^z commute. Hence we can deform the diagram by moving G_3^x one round backward and moving G_1^z one round forward. In addition, we can move G_1^x one round forward and move G_3^z one round backward. It gives an equivalent circuit:

$\cdots$	M^x		$G_1^xG_2^xG_3^x$		M^x		$G_1^xG_2^xG_3^x$	$\cdots$
$\cdots$		M^z		$G_1^zG_2^zG_3^z$		M^z		$\cdots$

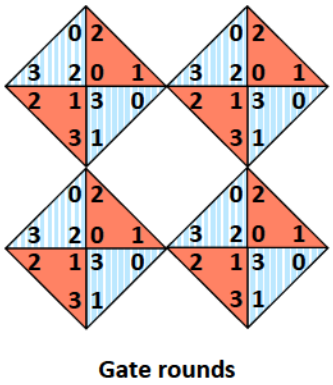


FIG. 10: Example of a correct syndrome extraction schedule with four rounds labeled by $0, 1, 2, 3$ repeated cyclically. The numbers assigned to vertices of each triangle indicate the rounds at which the code qubits comprising a triangle are coupled to the ancilla by CNOT gates. Measurement rounds are shown on Fig. 9. One can check that any pair of x-type and z-type triangles satisfies conditions of Lemma 1.

Our choice of a CNOT schedule is shown on Fig. 10. One can easily check that it satisfies conditions of Lemma 1. It remains to define the classical post-processing step that extracts the syndromes from the measured eigenvalues of triangle operators. For any integer $t \in [0, L-1]$ and a plaquette p we define a syndrome bit $s_p^Z(t)$ as a product of eigenvalues of Z-type triangles SE and NW located at the plaquette p that were measured at rounds $4t+1$ and $4t+2$. Similarly, we define a syndrome bit $s_p^X(t)$ as a product of eigenvalues of X-type triangles SW and NE located at the plaquette p that were measured at rounds $4t+3$ and $4t+4$. Hence each syndrome bit combines eigenvalues of two triangle operators measured in two consecutive rounds.

Let us now move to the error correction protocol that takes as input the syndrome information and outputs a correcting Pauli operator E^* acting on the code qubits. It mostly follows [3, 8, 27]. Our protocol deals with X-type and Z-type errors independently. It should be noted that the schedule shown on Figs. 9,10 is invariant under the horizontal reflection of the lattice and shifting the time by two rounds. Since the horizontal reflection exchanges X-type and Z-type triangles, it suffices to analyze X-type errors.

Let us introduce a 3D virtual lattice Λ that consists of virtual vertices and virtual edges. A virtual vertex is a

pair (p,t), where p is a plaquette of the 2D code lattice and $t \in [0, L-1]$ is the discrete time. We shall say that a virtual vertex $u = (p,t)$ has a *defect* iff the syndrome bits $s_p^Z(t)$ and $s_p^Z(t+1)$ are different. Hence the full syndrome history can be regarded as a configuration of defects on the virtual lattice.

We begin by considering configurations of defects created by a single fault in the readout circuit. Here a single fault includes one of the following possibilities:

- Wrong measurement outcome on some ancilla,
- Wrong ancilla preparation,
- One of the errors IX, XI, or XX inserted after some CNOT gate.

In other words, a single fault is any event that can occur with probability $\Omega(p)$ in the limit $p \to 0$ (recall that we only keep track of X-type errors). In order to define virtual edges we need the following observation.

Lemma 2. *Any single fault in the readout circuit creates either* 0 *or* 2 *defects on the virtual lattice.*

Proof. A measurement error on the ancilla a^z at a plaquette p creates two defects at virtual vertices (p,t) and $(p,t+1)$ for some t. Measurement errors on ancillas a^x create no defects since we ignore X-type syndromes. Ancilla preparation error can be regarded as an X-type error for ancillas a^z and Z-type error for ancillas a^x. Such an error can be propagated forward in time without feeding back to the code qubits because a^z is always a target qubit and a^x is always a control qubit for any CNOT gate, see Fig. 8. Hence ancilla preparation errors are equivalent to the measurement errors. By the same reason, an X-error on a^z caused by any CNOT gate is equivalent to a measurement error.

Consider now a single-qubit X error on some code qubit. If the syndrome were measured on all plaquettes directly after the error, one would observe non-trivial syndromes s_p^Z, s_q^Z at some pair of plaquettes p, q, see Section II. Since there are no other errors in the readout circuit, it will faithfully simulate the ideal syndrome measurements, that is, the syndrome $s_p^Z(t)$ will change from 1 to -1 for some step t_p and the syndrome $s_q^Z(t)$ will change from 1 to -1 for some step t_q. This produces a pair of defects at virtual sites (p,t_p) and (q,t_q). (More detailed analysis shows that either $t_p = t_q$ or $t_p = t_q \pm 1$). An error XX that occurred after a CNOT gate is equivalent to a single X error on the control qubit that occurred before this CNOT. The only remaining case is a single X error on the ancilla a^x. It can be propagated forward or backward towards the nearest a^x-measurement. Such propagation feeds back at most one X error to code qubits. This is the case that we have already explored. □

A trivial corollary of the lemma is that the number of defects on the virtual lattice is always even.

We connect a pair of virtual vertices u, v by a virtual edge iff there a single fault in the readout circuit capable of creating a pair of defects at u and v. A more detailed analysis shows that the virtual lattice has seven types of edges (not counting the orientation). The table below shows all 14 neighbors v of some fixed virtual vertex $u = (x,y,t)$ as well as the total number of single faults of each type in the readout circuit that create defects at u and v. For brevity we refer to single faults IX, XI, XX as G-faults (gate faults), while measurement and preparation single faults are referred to as M-faults and P-faults.

Neighbor of (x,y,t)	G-faults	M-faults	P-faults	Prior
$(x,y,t\pm 1)$	6	2	2	$11p/2$
$(x\pm 1,y,t)$	8	0	0	$2p$
$(x\pm 1,y\mp 1,t)$	8	0	0	$2p$
$(x,y\pm 1,t)$	4	0	0	p
$(x,y\pm 1,t\pm 1)$	2	0	0	$p/2$
$(x\mp 1,y,t\pm 1)$	2	0	0	$p/2$
$(x\mp 1,y\pm 1,t\pm 1)$	2	0	0	$p/2$

Note that the space-like virtual edges (those for which u and v have the same t coordinate) correspond to edges of the 2D virtual lattice defined in Section V. If one ignores the t coordinate, space-like virtual edges correspond to the code qubits. In particular, pairs of defects located on space-like virtual edges can be viewed as memory errors. On the other hand, pairs of defects located on time-like virtual edges(those for which u and v have the same x,y coordinates) can be viewed as syndrome measurement errors. The remaining virtual edges represent various combinations of memory errors and measurement errors.

For every virtual edge e we define a *prior* p_e as the probability to observe a pair of defects at the endpoints of e. Taking into account that any single G-fault has probability $p/4$, while a single M-fault and a single P-fault have probability p, one arrives at the priors listed in the table.

We shall choose the correction operator E^* by pretending that the creation of defect pairs on different virtual edges are independent events. Then the most likely combination of memory errors and measurement errors consistent with the observed configuration of defects coincides with the minimum weight matching of defects on the virtual lattice, where an edge e is assigned a weight $w_e \sim \log(1/p_e)$. The minimum weight matching M can be found efficiently using the Edmonds's algorithm. Finally, we choose the correction operator E^* as the product of all memory errors that appear in M. In order to decide whether the error correction is successful we compare E^* with the accumulated error E on the code qubits generated by the syndrome readout circuit. The results of our Monte Carlo simulation are shown on Figs. 11,12. It indicates that the threshold error rate for the circuit-based error model is $p_c \approx 0.6\%$.

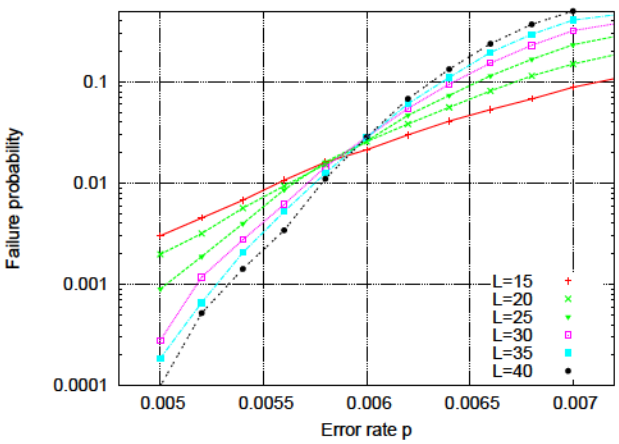


FIG. 11: Probability of the error correction failure for the subsystem toric code under the circuit-based error model. For a lattice of size L the syndrome measurement has been repeated L times. Each data point was obtained using $10^4 - 10^6$ Monte Carlo trials. The simulation was performed only for X-type errors (Z-type errors on the reflected lattice).

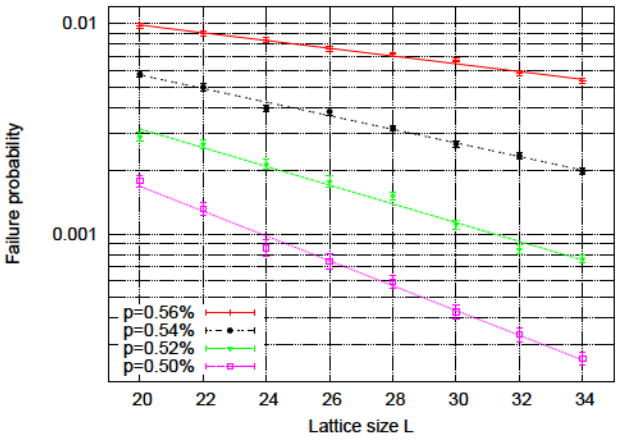


FIG. 12: Scaling of the error correction failure probability for different error rates below the threshold. Error bars represent the statistical error. Achieving the failure probability 10^{-6} at the error rate $p = 0.5\%$ would require $L \approx 100$.

VII. DIRECT 3-QUBIT PARITY MEASUREMENTS

In the previous section we estimated the threshold in a model where the triangle operators are measured by performing CNOT gates between the code qubits and an ancillary qubit. In this section, we will study a different model where measurements are performed directly. For some physical devices, it is possible to directly probe a multi-qubit parity operator without the need to use an ancillary qubit—the probe itself is used as a mediator that stores and accumulates the multi-qubit correlations. Direct two-qubit measurements have been realized in circuit quantum electrodynamics [28–31], and there are proposals [32] to turn these two-qubit measurements into parity measurements. These direct measurement schemes require a separate threshold analysis because they have different noise models and propagation.

We will focus on the recent proposal of DiVincenzo and Solgun [21] that realizes a three-qubit parity measurement ZZZ by capacitively coupling three Josephson-junction qubits to two transmission-line resonators. In the dispersive regime—where the difference between the resonant frequency of the transmission lines and the qubit transition frequency is much larger than the coupling strength—the transmission-line will pick-up a qubit-state-dependent frequency shift (Stark shift). When a near-resonant frequency probe signal is sent through one of the transmission lines, it picks up a phase that depends on the resonant frequency. Thus, the state-dependent resonant frequency shift will imprint a phase shift on the probe pulse that depends on the state of all three qubits. With an appropriate choice of parameters (qubit-transmission line detuning, coupling strength, probe signal frequency), the probe signals sent through the two transmission lines can be measured interferometrically to reveal information only about the parity of the three qubits, all other information about the qubit state imprinted on the probe signals being erased by the interferometric measurement. One important advantage of this measurement scheme is that a single qubit can participate to two distinct parity measurements simultaneously with an appropriate arrangement of transmission lines.

The main source of errors in this measurement is dephasing caused by the finite bandwidth of the probe pulse [21]. In an ideal parity measurement, the coherence between two computational basis states of the three-qubit systems $|x\rangle$ and $|y\rangle$ would be totally suppressed when x and y have different parities, and unaffected otherwise. The finite bandwidth of the probe pulse will cause dephasing between states of a given parity, and incomplete dephasing between states of distinct parity. Moreover, these errors are otherwise uniform, they do not depend, e.g., on the Hamming distance $|x-y|$ between the computational basis states. This is characteristic of a collective noise model, where multi-qubit errors are as likely as single qubit errors. In contrast, when qubits are subject to independent noise, dephasing would increase with Hamming distance between the states.

Parity measurements in the conjugate basis are required to measure the X-type triangle operators. These can be realized by rotating the qubits prior to sending the probe signal in the transmission line. Single qubit rotations are very fast and accurate in this architecture [33]. Nonetheless, they propagate errors, and can interchange X-type and Z-type errors. Based on these considerations, we will model noisy measurement of X-type and Z-type triangles in the following way:

- A noisy XXX or ZZZ measurement is modeled by a perfect even/odd subspace projection, followed by an error $(1-p)Id+p\mathcal{D}$, where Id is the identity map and $\mathcal{D}$ is the fully depolarizing three-qubit map ap-

plying one of 64 three-qubit Pauli operators with probability 1/64 each.

- The measurement outcome is flipped with probability p.

With this model, the syndrome extraction cycle requires only two rounds: one to measure all X-type triangles and one to measure all Z-type triangles. This implies that some qubits participate to two simultaneous measurements. As mentioned above, this is not a problem physically so far as a single qubit can be coupled to multiple transmission lines, which has already been demonstrated experimentally [34]. Moreover, two noisy parity measurements in the same basis always commute with our choice of noise model.

We have simulated fault-tolerant error correction of the subsystem toric code using direct parity measurement with the noise model described above, our results are presented on Fig. 13. Since errors are correlated in this model, we have opted for the renormalization group (RG) decoding algorithm proposed in [11, 12]. Indeed, Edmonds's minimum weight matching algorithm assumes an independent noise model and consequently yields a lower threshold in the presence of correlations. The RG decoder can exactly account for some of these correlations (those that are contained within a RG unit cell). Additional correlations can be approximated by updating the error prior on each RG unit cell using a belief propagation decoder [35]. The RG decoding is executed using these updated error priors. Following the proposal of [20], we map the code and its (updated) noise model onto the standard toric code on which we execute RG [38]. While Refs. [11, 12] assumed noiseless syndrome measurements, the decoding algorithm can easily be extended to noisy syndrome by renormalizing a 3D lattice [36].

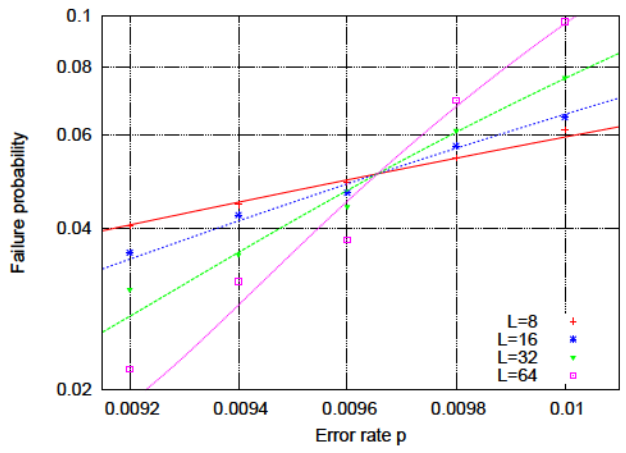


FIG. 13: Decoding error probability as a function of the physical error rate p for various lattice sizes L. The lines are obtained using the fitting functions $a + b(p - p_{th})L^{1/\nu} + c[(p - p_{th})L^{1/\nu}]^2$, which produces a threshold $p_{th} \approx 0.968\%$ and a critical exponent $\nu \approx 1.36$.

The results shown on Fig. 13 indicate a threshold value of roughly 0.97%. This value should be seen as lower bound to the true threshold of this code, which may well be above 1%. Indeed, the results presented here were obtained from a unit cell of dimension $2 \times 2 \times 1$: two of the three space-time dimensions are renormalized at each iteration, and by rotating the unit cell at each iteration we obtain a renormalization of the entire space-time lattice by a factor 4 after 3 RG rounds. Based on our experiments, larger RG unit cells produce higher thresholds because they make use of more correlations existing in the noise model. The decoding complexity scales exponentially with the unit cell size however, so in practice we are limited to relatively small cells. Furthermore, while the simulated syndrome extraction protocol involved measurements of both X-type and Z-type triangles, error correction has been performed only for X-type errors. Applying the same error correction algorithm independently to X-type and Z-type errors would result in the same error threshold [39]. We expect that more sophisticated decoders taking into account correlations between X and Z errors could achieve higher error thresholds.

Finally, we note that direct parity measurements of weight-four operators such as $ZZZZ$ and $XXXX$ can be realized similarly [21]. This could be used to implement Kitaev's toric code. However, it is not reasonable to assume that the noise rate p is independent of the weight w of the operator being measured. Thus, one needs to work out this dependence $p(w)$ from physical considerations before comparing thresholds of different codes obtained from direct parity measurements.

VIII. CONCLUSION

We have presented a subsystem version of Kitaev's surface code. The main features of our code is that it requires only 3-qubit parity measurements and its stabilizer generators have weight 6. Minimizing the weight of the parity measurements is helpful as it simplifies the measurement procedure, while minimizing the weight of the stabilizer generators is also desirable since it makes syndromes more reliable. In contrast to our code, the standard toric code requires weight 4 parity measurements and has weight 4 stabilizer generators. The subsystem color codes require only weight 2 parity measurements, but have stabilizer generators of weight up to 18. Thus, based only on these considerations, it is not clear how the threshold of these various codes should compare.

Our numerics show that in the circuit based model, our code has a threshold (0.6%) which is almost an order of magnitude larger than the one of the color code (0.08%) [37], and a bit more than half that of the standard toric code (0.9%) [5]. Motivated by the recent work of DiVincenzo and Solgun [21], we have also considered a setting where parity measurements can be implemented directly and found a threshold of 0.97%. This value cannot be

compared directly to the thresholds reported above since it is based on a substantially different noise model, and additional physical considerations must be taken into account before comparing.

We have shown that the new subsystem toric code gives rise to an exactly solvable spin Hamiltonian with 3-qubit interactions and topologically ordered ground state which is locally equivalent to the standard toric code.

IX. ACKNOWLEDGEMENTS

We would like to thank David DiVincenzo for helpful discussions and for drawing our attention to Ref. [21]. This work was supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract D11PC20167. S.B. was partially supported by DARPA QUEST program under contract number HR0011-09-C-0047. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Computational resources were provided by Calcul Québec and by IBM Blue Gene Watson supercomputing center. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

[1] A. Y. Kitaev, Annals of Physics **303**, 2 (2003).
[2] S. Bravyi and A. Y. Kitaev, ArXiv quant-ph/9811052 (1998).
[3] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[4] D. P. DiVincenzo, Physica Scripta Volume T **137**, 014020 (2009).
[5] A. Fowler, A. Whiteside, and L. Hollenberg, Phys. Rev. Lett. **108**, 180501 (2012).
[6] H. Bombin and M. A. Martin-Delgado, J. Phys. A: Math. Theor. **42**, 095302 (2009).
[7] R. Raussendorf and J. Harrington, Phys. Rev. Lett. **98**, 190504 (2007).
[8] A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A **80**, 052312 (2009).
[9] R. Raussendorf, J. Harrington, and K. Goyal, New J. Phys. **9**, 199 (2007), arXiv:quant-ph/0703143.
[10] A. G. Fowler, A. C. Whiteside, and L. C. Hollenberg, Phys. Rev. Lett. **108**, 180501 (2012).
[11] G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. **104**, 050504 (2010).
[12] G. Duclos-Cianci and D. Poulin, Information Theory Workshop (ITW), 2010 IEEE pp. 1–5 (2010).
[13] S. Bravyi and J. Haah, arXiv:1112.3252 (2011).
[14] H. Bombin, Phys. Rev. A **81** (2010).
[15] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97** (2006).
[16] D. Bacon, Phys. Rev. A **73**, 012340 (2006).
[17] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).
[18] S. Bravyi, Phys. Rev. A **83**, 012320 (2011).
[19] M. Suchara, S. Bravyi, and B. M. Terhal, J. Phys. A: Math. Theor. **44**, 155301 (2011).
[20] H. Bombin, G. Duclos-Cianci, and D. Poulin, arXiv:1103.4606 (2010).
[21] D. P. DiVincenzo and F. Solgun, arXiv:1205.1910 (2012).
[22] D. Aharonov and L. Eldar, arXiv:1102.0770 (2011).
[23] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
[24] X. Chen, Z.-C. Gu, and X.-G. Wen, Phys. Rev. B **82**, 155138 (2010).
[25] H. Bombin, arXiv:1107.2707 (2011).
[26] S. de Queiroz, Phys. Rev. B **73**, 064410 (2006).
[27] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Phys. Rev. A **83**, 020302(R) (2011).
[28] J. Chow, L. DiCarlo, J. Gambetta, A. Nunnenkamp, L. Bishop, L. Frunzio, M. Devoret, S. Girvin, and R. Schoelkopf, Phys. Rev. A **81**, 062325 (2010).
[29] S. Filipp, P. Maurer, P. Leek, M. Baur, R. Bianchetti, J. Fink, M. Goeppl, L. Steffen, J. Gambetta, A. Blais, et al., Phys. Rev. Lett. **102**, 200402 (2009).
[30] L. DiCarlo, J. Chow, J. Gambetta, L. Bishop, B. Johnson, D. Schuster, J. Majer, A. Blais, L. Frunzio, S. Girvin, et al., Nature **460**, 240 (2009).
[31] D. Riste, J. van Leeuwen, H.-S. Ku, W. Lehnert, and L. DiCarlo, arXiv:1204.2479 (2012).
[32] A. B. K. Lalumière, J.M. Gambetta, Phys. Rev. A **81**, 040301 (2010).
[33] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki, et al., arXiv:1203.4550 (2012).
[34] B. R. Johnson, M. D. Reed, A. A. Houck, D. I. Schuster, L. S. Bishop, E. Ginossar, J. M. Gambetta, L. DiCarlo, L. Frunzio, S. M. Girvin, et al., Nature Phys. **6**, 663 (2010).
[35] D. Poulin and Y. Chung, Quant. Inf. and Comp. **8**, 986 (2008).
[36] G. Duclos-Cianci and D. Poulin, In preparation.
[37] A. J. Landahl, J. T. Anderson, and P. R. Rice, arXiv:1108.5738 (2011).
[38] This last step plays no fundamental role, it only saves us from programming a distinct RG decoder for each code: since the decoding problem of any 2D translationally-invariant stabilizer code can be mapped to that of decoding of the standard toric code [20], the same program can be used with different codes.
[39] Since the subsystem toric code and the error model are symmetric under exchange of Pauli X and Z, the error threshold must be the same for the models with only X and only Z errors. Correcting each type of errors independently for the full error model can increase the decoding failure probability at most by a factor of two compared with the case of X errors only (use the union bound).

Conclusion

Dans cette thèse, j'ai présenté l'ensemble de mes travaux qui ont touché deux aspects importants du calcul tolérant aux fautes. Premièrement, j'ai travaillé sur des protocoles permettant de réaliser un ensemble universel de portes de manière tolérante aux fautes. Deux d'entre eux sont basés sur l'injection d'états magiques. Avec Krysta Svore, j'ai découvert une nouvelle famille d'états que nous avons appelée « échelle d'états » (Chapitre 2). J'ai proposé le circuit permettant de les créer et j'ai montré qu'il effectuait en réalité une faible distillation. J'ai aussi revisité la notion de compilation dans le but d'utiliser ces états ressources. J'ai ensuite observé des gains par rapport aux protocoles déjà établis à ce moment. Plus tard, j'ai poussé ces travaux avec David Poulin en étudiant une autre famille d'états magiques, plus élégante, ce qui nous a permis de mieux traiter le problème de la compilation (Chapitre 3). J'ai conçu et analysé les circuits qui en permettent la distillation. Encore une fois, j'ai observé des gains pour certaines rotations de Pauli qui sont pertinentes à la transformée de Fourier quantique et à la simulation en chimie quantique. Le troisième protocole élaboré avec Jonas Anderson et David Poulin profite d'une autre approche qui ne nécessite pas d'états magiques (Chapitre 4). Il s'agit d'une déformation de codes tolérante aux fautes. Nous avons montré qu'à l'aide de seulement quinze qubits, il est possible de réaliser un ensemble universel de portes en déformant le code Steane en code de Reed-Muller et vice-versa.

Deuxièmement, j'ai travaillé à généraliser la méthode de décodage de codes topologiques par renormalisation que j'ai développée au cours de ma maîtrise. Tout d'abord, j'ai collaboré avec Héctor Bombin et David Poulin pour montrer que tous les codes topologiques stabilisateurs invariants sous translation sont équivalents, c.-à-d. qu'ils appartiennent tous à la même phase topologique (Chapitre 5). Cette équivalence se traduit de manière opérationnelle : il existe une transformation de Clifford locale permettant de passer d'un code à l'autre. Cette transformation permet alors d'adapter une méthode de décodage sur un code à tous les autres de manière naturelle. Cela m'a permis d'utiliser mon décodeur du code topologique de Kitaev pour décoder le code de couleurs 4.8.8 et le code à sous-systèmes associé. Puis, j'ai adapté la méthode aux codes topologiques de Kitaev sur des qudits (Chapitre 6). Dans ce cas,

j'ai observé un accord inattendu entre les seuils obtenus et la borne de hachage généralisée. Toutefois, cela est cohérent avec la conjecture de Nishimori. Ensuite, j'ai généralisé mon décodeur au cas tolérant aux fautes et j'ai observé un seuil comparable aux méthodes de décodages par appariement habituellement utilisées (Chapitre 7). Aussi, j'ai appliqué la méthode au code de surface à sous-systèmes de Sergey Bravyi et j'ai montré que dans le cas où les mesures de syndrome à trois qubits sont effectuées de manière simultanées, le seuil observé est comparable au seuil du code topologique de Kitaev habituel (Chapitre 8).

Finalement, énumérons quelques pistes de recherches intéressantes donnant suite aux articles présentés ci-haut. Dans le but de comparer les différentes méthodes de compilation et de distillation, il serait judicieux qu'une seule équipe simule tous les protocoles pour une collection d'angles donnés. Le problème, c'est que chacun des protocoles a été étudié en vase clos, en prenant des moyennes, en faisant des approximations, etc. Il est alors difficile de comparer de manière juste les différents protocoles. Par rapport aux travaux effectués avec Krysta Svore, une étude plus systématique du processus de « densification » de l'échelle d'états à l'aide de nouveaux circuits de Clifford serait profitable. Par exemple, étant donné un nombre m de qubits dans l'état $|H\rangle$ et un nombre n de qubits stabilisateurs, que peut-on dire des états qu'il est possible de créer à l'aide d'un circuit de Clifford et de mesures de Pauli ? D'un autre point de vue, cela revient à généraliser le circuit d'injection. Par rapport aux travaux sur l'autre protocole de distillation élaboré avec David Poulin, il serait profitable d'optimiser le programme (*schedule*) de distillation. Aussi, dans la deuxième partie du même article, portant sur le protocole généralisé, l'analyse a été faite de manière approximative à l'aide de courbes ajustées aux données. Il serait bon de pousser aussi loin que possible l'analyse exacte des erreurs. Encore une fois, il est possible de densifier la famille d'états accessibles comme discuté dans l'article. Il faudrait analyser les performances de distillation pour ces états. Les travaux sur la déformation de codes nous ont amenés à considérer les codes de Reed-Muller à rendement constant. Quelles sont les portes transverses de ces codes ? Permettent-elles de meilleurs rendements de distillation d'états magiques ? Enfin, il serait intéressant de développer et de programmer un standard efficace permettant d'interfacer le problème du décodage sur n'importe quel code topologique stabilisateur au décodeur du code topologique de Kitaev, tolérant aux fautes ou non.

Bibliographie

[1] Tomas Jochym-O'Connor et Raymond Laflamme. *Physical Review Letters* **112**(1), 010505 (2014).

[2] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, et R. J. Schoelkopf. *Physical Review A* **76**(4) (2007).

[3] Daniel Gottesman. *Physical Review A* **57**(1), 127–137 (1998).

[4] Eric Dennis, Alexei Kitaev, Andrew Landahl, et John Preskill. *Journal of Mathematical Physics* **43**(9), 4452–4505 (2002).

[5] Martin Suchara, Arvin Faruque, Ching-Yi Lai, Gerardo Paz, Frederic T Chong, et John Kubiatowicz. *arXiv preprint arXiv:1312.2316* (2013).

[6] Christopher M Dawson et Michael A Nielsen. *arXiv preprint quant-ph/0505030* (2005).

[7] Vadym Kliuchnikov, Dmitri Maslov, et Michele Mosca. *Quantum Information & Computation* **13**(7-8), 607–630 (2013).

[8] Adam Paetznick et Krysta M Svore. *Quantum Information & Computation* **14**(15-16), 1277–1301 (2014).

[9] Neil J Ross et Peter Selinger. *arXiv preprint arXiv:1403.2975* (2014).

[10] Michael A Nielsen et Isaac L Chuang. *Quantum computation and quantum information*. Cambridge university press, (2010).

[11] John Preskill. *California Institute of Technology* (1998).

[12] D Gottesman. *Stabilizer codes and quantum error correction*. Thèse de Doctorat, Caltech, (1997).

[13] Guillaume Duclos-Cianci. Mémoire de Maîtrise, Université de Sherbrooke, (2010).

[14] Alexander Yu. Vlasov. *Physical Review A* **63**(5) (2001).

[15] Hector Bombin et Miguel Angel Martin-Delgado. *Physical Review Letters* **97**(18), 180501 (2006).

[16] Sergey Bravyi et Alexei Kitaev. *Physical Review A* **71**(2), 022316 (2005).

[17] Ben W Reichardt. *Quantum Information Processing* **4**(3), 251–264 (2005).

[18] Charles H Bennett, David P DiVincenzo, John A Smolin, et William K Wootters. *Physical Review A* **54**(5), 3824 (1996).

[19] Raymond Laflamme, Cesar Miquel, Juan Pablo Paz, et Wojciech Hubert Zurek. *Physical Review Letters* **77**(1), 198 (1996).

[20] Andrew M Steane. *Physical Review Letters* **77**(5), 793 (1996).

[21] Alex Bocharov, Yuri Gurevich, et Krysta M Svore. *Physical Review A* **88**(1), 012313 (2013).

[22] Krysta Svore et Guillaume Duclos-Cianci (2013). US Patent App. 13/948,171.

[23] Scott Aaronson et Daniel Gottesman. *Physical Review A* **70**(5), 052328 (2004).

[24] Andrew J Landahl et Chris Cesare. *arXiv preprint arXiv:1302.3240* (2013).

[25] Adam M Meier, Bryan Eastin, et Emanuel Knill. *arXiv preprint arXiv:1204.4221* (2012).

[26] H Bombin. *Physical Review A* **81**(3), 032301 (2010).

[27] H Bombin et MA Martin-Delgado. *Physical Review A* **77**(4), 042322 (2008).

[28] A Yu Kitaev. *Annals of Physics* **303**(1), 2–30 (2003).

[29] Hidetoshi Nishimori et Koji Nemoto. *Journal of the Physical Society of Japan* **71**(4), 1198–1199 (2002).

[30] Daniel Gottesman. Dans *Quantum Computing and Quantum Communications*, 302–313. Springer (1999).

[31] David P DiVincenzo et Firat Solgun. *New Journal of Physics* **15**(7), 075001 (2013).